BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Evaluating the Robustness of Image Steganography Techniques against Diverse Degradation Techniques

Lamiya Tahsin Orpa
190041208
Rahanuma Ryaan Ferdous
190041228
Umme Tasnim Hasan
190041241

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Evaluating the Robustness of Image Steganography Techniques against Diverse Degradation Techniques

Lamiya Tahsin Orpa
190041208
Rahanuma Ryaan Ferdous
190041228
Umme Tasnim Hasan
190041241

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Lamiya Tahsin Orpa**, **Rahanuma Ryaan Ferdous**, and **Umme Tasnim Hasan** under the supervision of **Dr. Md. Hasanul Kabir**, Professor and Head of the Department, Department of Computer Science and Engineering and co-supervision of **Shahriar Ivan**, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Dr. Md. Hasanul Kabir
Professor and Head of the Department
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 04, 2024

Shahriar Ivan
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 04, 2024

Lamiya Tahsin Orpa
Student ID: 190041208
Date: June 04, 2024

Rahanuma Ryaan Ferdous
Student ID: 190041228
Date: June 04, 2024

Umme Tasnim Hasan
Student ID: 190041241
Date: June 04, 2024

Contents

List of Figures

List of Tables

Abstract

Within the field of steganography, image steganography is a fascinating technique that hides confidential data behind a cover image. However, the quality and security of hidden content are threatened by potential degradation introduced by the transmission of stego images, which can take the form of noise, blurring, or sharpening. In this research endeavor, we present an innovative approach by integrating image degradation models into an existing state-of-the-art invertible image-in-image steganography model known as HiNet. Our proposed methodology involves applying a degradation model to the stego image post-secret image embedding, followed by the usual secret image extraction. By introducing these additional layers to the steganographic process, we aim to enhance the robustness of our model against various degradation scenarios, such as Gaussian noise, blurring, and sharpening. By using knowledge from cutting-edge architectures such as HiNet, we aim to improve the overall security and quality of hidden pictures. We performed experiments on HiNet and achieved improved results in handling degradations like noise, blurring, and sharpening.

Chapter 1

Introduction

1.1 Overview

Information hiding can be done with many methods. Steganography is one of them. “Steganography” is derived from the Greek words “steganos” meaning “covered” and “graphein” meaning “writing”, which translates to “covered writing” or “hidden writing”. It is a method of concealing secret data within a cover object so that the existence of the data is not revealed to an unauthorized observer. The cover object can be any digital media, such as text, audio, video, or image. The secret data can be any type of information, such as text, QR code, string, hyperlink, or image. The goal of steganography is to achieve obscurity, which means that the hidden data is not detectable by any means, either visually or statistically. Steganography can be of different types: Image, Text, Audio, Video. Image steganography is a special case of steganography, where the cover object and the secret data are both images. Image steganography has many advantages over other types of steganography, such as:

- Images are widely used and exchanged on the internet, which makes them suitable for covert communication.
- Images have high redundancy and capacity, which means that they can store a large amount of data without noticeable distortion.
- Images have various formats and properties, such as color, size, resolution, compression, and noise, which can be exploited for hiding data in different ways.

Image steganography has various applications in different domains, such as:

- **Cyber security:** Image steganography can be used to protect sensitive or confidential information from unauthorized access or disclosure, such as passwords,

keys, credentials, or personal data.

- **Digital forensics:** Image steganography can be used to hide evidence or clues within images, such as timestamps, locations, identities, or messages, which can be useful for investigation or verification purposes.
- **Intelligence:** Image steganography can be used to transmit secret or classified information between agents or organizations, such as military, espionage, or terrorism, without raising suspicion or alerting the enemy.
- **Multimedia communication:** Image steganography can be used to enhance the functionality or quality of multimedia services, such as authentication, watermarking, annotation, or compression.

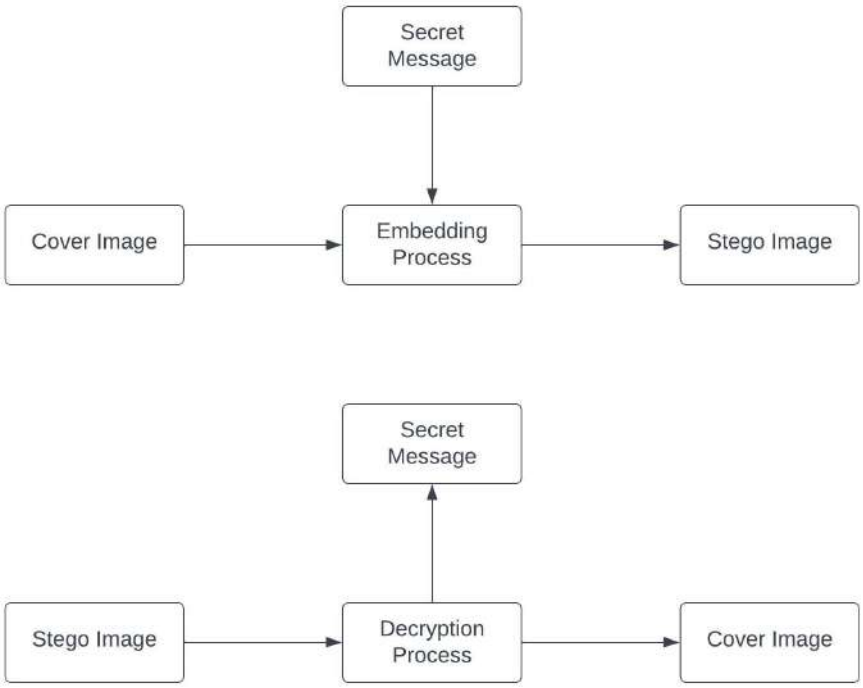


Figure 1.1: Basic flowchart of Steganography: (a) Embedding process and (b) Decryption process

Before going in depth, we need to understand some terminologies, which are used for understanding the process of Steganography. This section covers those terms:

- **Cover Object:** The cover object, also known as the **carrier** or **container object**, refers to the original image that conceals the secret information.
- **Secret Object:** The secret object, often referred to as the message, is the infor-

mation that needs to be concealed within the cover object. This can be text, an image, an audio or any data that the sender wants to keep confidential or hidden from unauthorized users.

- **Stego Object:** The stego object is the result of embedding the secret object within the cover object, creating a new composite file. The process of steganography aims to make this stego object indistinguishable from the original cover object to maintain covert communication and prevent detection by unintended recipients.
- **Embedding:** The process of hiding the secret message in the cover image.
- **Extraction:** The process of retrieving the secret message from the stego image.
- **Steganalysis:** The process of detecting the presence of hidden information in an image. It can be classified into two types: passive and active. Passive steganalysis only analyzes the image without modifying it, while active steganalysis tries to alter or destroy the hidden information.

1.2 Motivation and Scope

We opted for image steganography primarily due to its expansive scope and superior payload capacity. Through an exploration of various image steganographic papers, it became evident that HiNet [18] consistently delivers more favorable outcomes in comparison to other methods. Despite significant advancements in this field, the robustness of these techniques against various forms of image degradation remains a critical concern. While HiNet [18] has demonstrated cutting-edge performance, it does have some drawbacks that we have identified from the follow-up literature [52]. If the stego image undergoes degradation or becomes noisy during network transmission, recovery of the secret image is compromised; in some instances, it fails entirely. To illustrate this, let us see the example of recovered images that were subjected to various degradations in Figure 1.2:

On the left, the first column is the ground truth which represent the original image. The next column is our first type of degradation, which is JPEG compression with a quality factor of 90. In this case, the recovered image is completely unrecognizable. The next column shows the effect of degradation by Gaussian noise with a degradation level of $sigma = 10$, resulting in jittery and unrecognizable image.

On the right, we have different type of degradation involving social media apps like WeChat and Shoot. Sharing stego images via these apps leads to poor image recovery.

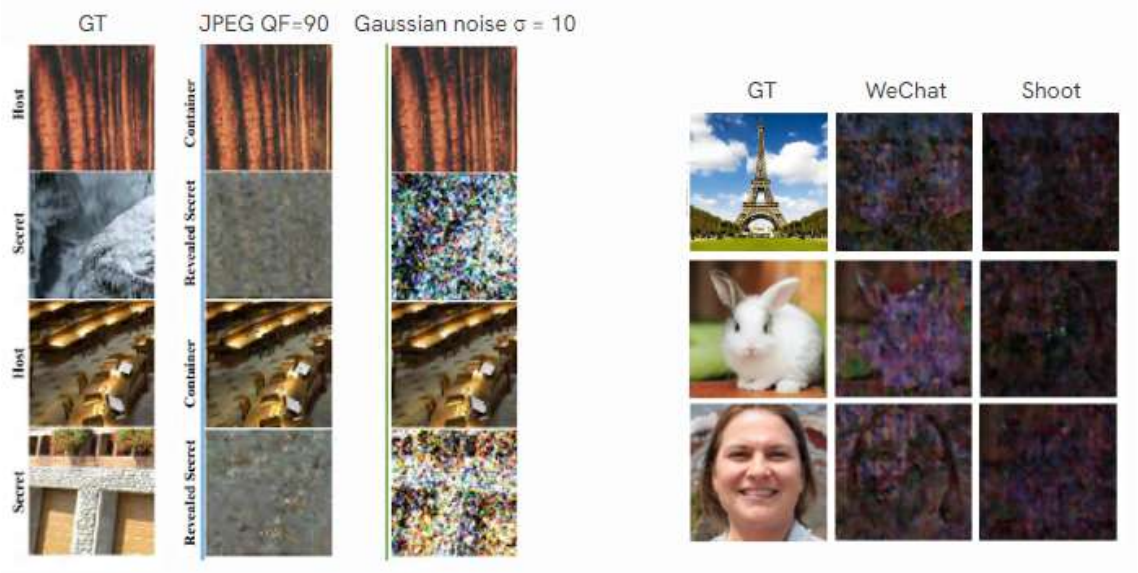


Figure 1.2: Visual comparisons of HiNet under real-world degradations

From CROSS we got to know some of the limitations of HiNet which is that it cannot handle any noise. So we decide to test this and also experimented whether this statement is true for other degradations or not. So we decided to test the SoTA HiNet on Gaussian noise, blurring and sharpening as well.

1.2.1 Robustness of HiNet towards Different Degradations

Gaussian Noise

The current iteration of HiNet faces challenges in mitigating external Gaussian noise. Despite undergoing the training process, HiNet struggles to decode the secret from distorted stego images corrupted with Gaussian noise. Even after the backward pass, HiNet remains incapable of successfully retrieving the secret from these perturbed stego images.

Testing Results: Upon conducting tests, HiNet exhibited limitations in handling Gaussian noise-corrupted stego images, failing to recover the secret information effectively.

Blurring

When subjected to blurring, the current HiNet model failed to produce the anticipated results. Instead of preserving the secret information, the blurring process led to significant degradation and loss of the secret within the stego images.

Testing Results: Upon evaluation, HiNet demonstrated an inability to effectively

counteract the effects of blurring, resulting in compromised secret information within the stego images.

Sharpening

During the evaluation with sharpening, notable contrasts were observed in the recovered secret information by HiNet.

Testing Results: HiNet showcased substantial discrepancies in the recovered secret information when subjected to sharpening. The sharpening process elicited stark contrasts in the retrieved secret, highlighting potential challenges in maintaining consistency and fidelity in secret recovery.

As our tests with various degradation techniques did not yield the expected results for HiNet, we decided to address this issue. We learned from StegaStamp that a model can handle real-world perturbations effectively if it is trained with those specific perturbations. Therefore, our approach was to improve HiNet's ability to handle degradations by training the existing state-of-the-art model with these degradations. After completing the training, we aimed to assess whether the model could properly retrieve the secret even when the stego image is distorted.

In our investigation on how to improve the robustness of our main paper, HiNet [18], we were inspired by the concept of introducing a degradation process as mentioned by Tancik et al. [39] in between the concealing network and the revealing network.

1.3 Problem Statement

Although HiNet [18] serves as a robust steganographic network, excelling in the effective hiding and concealing of secret images, recent studies indicate its susceptibility to noise. Real-world degradation poses a challenge, as it results in damage to both the stego images and the retrieved secret, hindering the primary objective of steganography. Unlike papers like RIIS and CROSS, which adeptly handle noise through diffusion models, limitations persist in CRoSS, particularly in pixel-wise objective fidelity metrics like PSNR.

In response, our proposal recommends the integration of a degradation model to augment the capabilities of HiNet [18]. This integration is intended to fortify the steganographic network, ensuring enhanced performance in scenarios with various forms of interference and noise.

1.4 Research Challenges

In an increasingly digital world, the importance of secure communication cannot be overstated. Image steganography, the art of hiding information within images, provides a covert channel for secure data transmission. The robustness of image steganography techniques can be examined through a comprehensive analysis of several key factors:

1. **Imperceptibility:** The degree to which the hidden data is invisible to the human eye or any automated analysis, such as steganalysis or distortion.
2. **Robustness against distortion:** The resistance of the hidden data to various attacks or modifications, such as cropping, scaling, filtering, or encryption.
3. **Security against steganalysis:** The degree of resistance against steganalysis. It depends on the statistical and perceptual properties of the cover and stego images, as well as the complexity of the embedding and extraction algorithms.
4. **Reliable extraction:** The ability to recover the hidden data from the image without errors or losses, even in the presence of noise, compression, or manipulation.
5. **Payload capacity:** The amount of data that can be hidden within an image without affecting its quality or size.

1.5 Contribution

The contributions of this thesis are centered on enhancing the robustness and performance of the HiNet model in handling various forms of image degradation. This section details the specific strategies implemented and their significance in advancing the field of image steganography.

- proposing a novel architecture that enhances robustness of the original HiNET architecture by doing so and so
- a thorough qualitative and quantitative analysis reflecting this and this fact that you wanted to test
- **Training HiNet with Noisy Stego**

  This approach significantly enhances HiNet's capability to handle and recover secret information from stego images affected by Gaussian noise, a common real-world degradation.

- **Training HiNet with Blurred Stego**

  By training with blurred images, we improved HiNet's performance in scenarios where image quality is compromised by blurring, such as when images are shared over platforms that may introduce such degradations.

- **Training HiNet with Sharpened Stego**

  This approach enhances HiNet's robustness against sharpening, ensuring that secret information can be reliably recovered even when stego images are subjected to this type of degradation.

The enhanced HiNet model exhibits superior robustness against various forms of image degradation, ensuring the reliable recovery of secret information from degraded stego images. This improvement is crucial for practical applications where images often undergo quality alterations. By addressing real-world perturbations such as noise, blurring, and sharpening, our modified HiNet model is more suited for use in environments where images are frequently subjected to these types of degradations, such as social media platforms and network transmissions. Our work provides a foundation for further research in enhancing the robustness of steganographic models. These advancements are vital steps forward in the field of image steganography, ensuring more reliable and secure communication in environments where image quality cannot be guaranteed.

1.6 Organization

This thesis is structured to guide the reader through the process of understanding the enhancements made to the HiNet model for improved robustness against image degradations. Each chapter systematically addresses different aspects of the research, from foundational concepts to experimental results. The organization of the thesis is as follows:

Chapter 1 provides an overview of the research, including the motivation and scope of the study. It introduces the HiNet model and discusses its limitations when exposed to various image degradations. The chapter also outlines the problem statement, research challenges, and the specific contributions of this thesis. It sets the stage for the detailed investigations and enhancements presented in subsequent chapters.

Chapter 2 reviews the literature on image steganography, cryptography, watermarking, and fingerprinting, comparing the HiNet model with other state-of-the-art methods. It includes a discussion on benchmarking datasets and evaluation metrics used

in the field. Additionally, the chapter covers classical and statistical methodologies, deep learning approaches, and generative techniques relevant to the research. Specific attention is given to prior works like SteganoGAN, StegaStamp, IDEAS, CRoSS, and WINNet.

Chapter 3 details the methodology adopted to enhance the HiNet model. It describes the overall approach, including the integration of degradation-aware training into the HiNet framework. The dataset used for training and evaluation is discussed, along with the metrics employed to assess performance. The proposed pipeline and specific modifications to the HiNet model, such as training with noisy, blurred, and sharpened stego images, are elaborated upon.

Chapter 4 presents the experimental results obtained from testing the enhanced HiNet model. It includes a discussion on the hyperparameters tuned during the experiments and the outcomes of training with various image degradations. The chapter provides a qualitative and quantitative analysis of the results, comparing the performance of the enhanced HiNet model against its baseline version. Visual and statistical comparisons illustrate the improvements achieved.

We see chapter 5 summarizing the research findings, highlighting the significance of the contributions made. It discusses the implications of the enhanced HiNet model for practical applications and suggests directions for future research. The chapter reinforces the importance of robust image steganography in real-world scenarios and the potential for further advancements in this field.

Each chapter is designed to build upon the previous one, ensuring a logical flow of information that seamlessly guides the reader from the introduction to the detailed methodology, results, and finally, the conclusion. This structured approach aims to provide a clear and comprehensive understanding of the enhancements made to the HiNet model and their impact on the field of image steganography.

Chapter 2

Related Works

2.1 Comparison with related methods

2.1.1 Cryptography

Steganography is different from cryptography, which is the study and practice of making data unreadable or unrecognizable by using mathematical transformations. Cryptography hides the meaning of the data, while steganography hides the existence of the data. Cryptography can be easily detected by observing the ciphertext, which is the encrypted data, while steganography can be more stealthy by embedding the data within a seemingly innocent object. However, cryptography can provide more security by making the data incomprehensible even if it is detected, while steganography can be vulnerable to extraction or modification if the cover object is compromised. Therefore, steganography and cryptography can be used in combination to achieve both security and obscurity by first encrypting the message and then hiding it in another medium. The medium can be text, image, audio or video. We are particularly focusing on image steganography.

2.1.2 Watermarking

Watermarking is the process of embedding a mark or logo within an image, such that it is perceptible or imperceptible to the human eye or any automated analysis. The mark or logo can be any type of information, such as text, image, or symbol. The image that contains the mark or logo is called the watermarked image. The goal of watermarking is to achieve authentication, which means that the mark or the logo can be used to verify the ownership, the origin, or the integrity of the image.

Watermarking is similar to image steganography in the sense that both methods hide information within an image. However, watermarking is different from image steganography in the following aspects:

- The information that is hidden by watermarking is usually related to the image itself, such as the author, the date, or the source, while the information that is hidden by image steganography can be unrelated to the image, such as a message, a link, or a code.
- The information that is hidden by watermarking is usually intended to be detected or extracted by anyone who has access to the image, such as the public, the customers, or the authorities, while the information that is hidden by image steganography is usually intended to be detected or extracted only by the authorized recipient, such as the sender, the receiver, or the ally.
- The information that is hidden by watermarking is usually visible or noticeable to the human eye or any automated analysis, such as a logo, a signature, or a pattern, while the information that is hidden by image steganography is usually invisible or undetectable to the human eye or any automated analysis, such as a pixel, a bit, or a noise.

Watermarking has various applications in different domains, such as digital rights management, content authentication, and tamper detection.

2.1.3 Fingerprinting

Fingerprinting is the process of embedding a unique or distinctive mark or code within an image such that it is imperceptible to the human eye or any automated analysis. The mark or the code can be any type of information, such as a number, a string, or a symbol. The image that contains the mark or the code is called the fingerprinted image. The goal of fingerprinting is to achieve identification, which means that the mark or the code can be used to trace the origin, the distribution, or the usage of the image.

Fingerprinting is similar to image steganography in the sense that both methods hide information within an image. However, fingerprinting is different from image steganography in the following aspects:

- The information that is hidden by fingerprinting is usually unique or distinctive for each image or each user, such as a serial number, a hash value, or a signature, while the information that is hidden by image steganography can be common

or shared for multiple images or users, such as a message, a link, or a code.

- The information that is hidden by fingerprinting is usually intended to be detected or extracted by the authorized owner or the provider of the image, such as the creator, the publisher, or the distributor, while the information that is hidden by image steganography is usually intended to be detected or extracted by the authorized recipient of the image, such as the sender, the receiver, or the ally.
- The information that is hidden by fingerprinting is usually invisible or undetectable to the human eye or any automated analysis, such as a pixel, a bit, or a noise, while the information that is hidden by image steganography can be visible or noticeable to the human eye or any automated analysis, such as a logo, a signature, or a pattern.

Fingerprinting has various applications in different domains, such as digital piracy prevention, content tracing, and user behavior analysis.

2.2 Benchmarking datasets

The success and efficacy of steganographic techniques are often evaluated and benchmarked against diverse datasets, each designed to address specific aspects of image processing, analysis, and steganalysis. In this section, we provide a concise overview of several prominent datasets employed in the field of steganography.

BOSSBase: BOSSBase, short for Break Our Steganographic System Base, is a collection of 10,000 grayscale images each sized at 512 x 512 pixels. Primarily designed for steganalysis research, it serves as a valuable resource for benchmarking and evaluating steganalysis algorithms, developing new steganographic techniques, and studying the impact of embedding hidden data on image features. The key features include diverse content covering various subjects and textures, absence of pre-embedded data (clean images without hidden messages), and its availability for non-commercial research purposes, providing researchers with a freely accessible dataset for their investigations.

BOSSRank: BOSSRank, standing for Break Our Steganographic System Rank, is a dataset consisting of 1,000 grayscale images in PGM (Portable Gray Map) format, each sized at 512 x 512 pixels. Specifically designed for evaluating steganalysis algorithms, its primary purposes include benchmarking and ranking steganalysis algorithms within a challenging context. The dataset serves to assess the robustness of

algorithms against various embedding techniques, providing a valuable resource for researchers and practitioners in the field.

CelebA: CelebA is a large-scale dataset featuring over 200,000 celebrity images, providing rich annotations with 40 binary attributes (e.g., hair color, gender). It includes landmark locations for eyes, nose, and mouth, covers a diverse range of poses and backgrounds, and features images of 10,177 different celebrities. The dataset is valuable for research in computer vision, facial recognition, and attribute analysis.

ImageNet: ImageNet is a massive dataset of over 14 million hand-annotated images, organized according to the WordNet hierarchy. It has played a pivotal role in advancing the field of computer vision, serving as a benchmark for image classification and object detection algorithms.

COCO: COCO (Common Objects in Context) is a large-scale dataset tailored for challenging computer vision tasks like object detection, segmentation, and captioning. With over 330,000 images, including more than 200,000 labeled ones, COCO offers a diverse array of complex everyday scenes. The dataset is marked by its extensive annotations, featuring multiple annotations per image, including bounding boxes for up to 150 objects across 80 categories, segmentation masks for individual objects, and five natural language captions describing each scene.

Div2K: One popular dataset designed for single-image super-resolution (SISR) research is DIV2K. It consists of one thousand high-quality photos covering a wide range of degradation types and content, making it an ideal tool for testing and training SISR algorithms. The dataset has a high resolution of 2K and is thoughtfully split into 800 training, 100 validation, and 100 testing photos. DIV2K facilitates the development and evaluation of algorithms aimed at enhancing image resolution from low-resolution inputs.

LSUN: The LSUN (Large-scale Scene Understanding Network) dataset is a compilation of diverse, high-resolution scene datasets designed for tasks such as object detection, scene understanding, and visual recognition. Although not a singular dataset, LSUN consists of multiple sub-datasets, each concentrating on a specific scene category, such as bedrooms, kitchens, streets, or horses. This extensive collection encompasses thousands of high-resolution images (256x256 pixels) within each sub-dataset, with over 2 million images collectively. Notably, LSUN provides realistic scenes, capturing the complexities of real-world environments, including variations in lighting, composition, and object arrangements.

Pascal VOC: The PASCAL VOC (Visual Object Classes) dataset is extensively em-

ployed for tasks like object detection, image segmentation, and classification. It includes images portraying everyday objects in real-world scenes and is characterized by detailed annotations encompassing object classes, bounding boxes, and pixel-level segmentation masks.

SIDD: Using five typical smartphone cameras, 30,000 noisy photos from ten scenarios in various lighting situations make up the SIDD image denoising dataset. The noisy photos are accompanied with ground truth images.

DND: Darmstadt Noise Dataset. 50 pairs of noisy and (almost) noise-free photos taken with four consumer cameras make up this dataset. The providers extract 20 crops, each measuring 512×512, from each image due to the high resolution of the photographs, resulting in a total of 1000 patches.

The following table provides an overview of all the used datasets in steganography:

Table 2.1: Overview of Datasets in Steganography

Dataset	**Number of Samples**	**Format**	**Size**	**Purpose**
BOSSBase BOSSRank	9074 (training) 1000 (testing)	PGM	512 x 512 x 1	Steganography and steganalysis
CelebA	More than 200K	TIFF	178 x 218 x 3	Face attribute recognition, face recognition, face detection, landmark localization, and face editing & synthesis
ImageNet	More than 14M	Arbitrary	Arbitrary	Computer vision, image processing
COCO	330,000	JPG	640 x 640 x 3	Object detection, segmentation, and image captioning
Div2K	120K - 3000K (training) 300 (validation) 1000 (testing)	PNG	1020 x 678 x 3	Single Image Super-Resolution
LSUN	800 (training) 100 (validation) 100 (testing)	JPG	256 x 256 x 3	Scene classification and understanding
Pascal VOC	11,530	JPG	500 x 300 x 3	Object detection and classification

2.3 Evaluation Metrics

- **PSNR (Peak Signal-to-Noise Ratio)**: Evaluates the stego-picture's quality by contrasting it with the original cover image. Higher PSNR values indicate less distortion.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{2.1}$$

- **SSIM (Structural Similarity Index Measure)**: Assesses the visual impact of changes in luminance, contrast, and structure between the cover and stego-images.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.2}$$

- **MSE (Mean Square Error)**: Quantifies the error between the cover and stego-images on a pixel-by-pixel basis.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i, j) - K(i, j)]^2 \tag{2.3}$$

- **LPIPS (Learned Perceptual Image Patch Similarity)**: Computes the similarity between the activations of two image patches for some pre-defined network that match human perception well.

$$LPIPS(x, y) = \frac{1}{N}\sum_{i=1}^{N} \|\phi_i(x) - \phi_i(y)\|_2 \tag{2.4}$$

  where x and y are image patches, N is the number of layers in the network, ϕ_i is the feature map of the i-th layer, and $\|.\|_2$ is the L2 norm.

- **BPP (Bits Per Pixel)**: Measures the payload capacity, indicating the amount of secret information embedded per pixel of the cover image.

$$BPP = \frac{\text{Number of Secret Bits Embedded}}{\text{Total Number of Pixels}} \tag{2.5}$$

- **Hiding Capacity**: Refers to the maximum amount of information that can be hidden within an image.

- **Correlation**: Evaluates the linear relationship between the pixel values of the original and stego-images.

- **Security Metrics**: Assesses the security aspect of steganography, like resistance to steganalysis attacks e.g. higher AUROC values indicate better detection accuracy.

2.4 Used Methodologies

2.4.1 Classical/Statistical

Statistical image steganography techniques usually come before neural network technology. To encode a secret message, Least-Significant Bit (LSB) techniques alter each pixel's lower-order bits. These techniques are quite simple and lossless, but they can be readily identified. To avoid being discovered by these steganalysis algorithms, numerous statistical image steganography techniques were put forth.

Highly Undetectable Steganography (HUGO) **(Pevný et al. 2010)** [33] employs handcrafted features to measure the distortion caused by pixel modifications and selects pixels that result in minimal distortion, making it one of the most secure steganographic techniques. Wavelet Obtained Weights (WOW) **(Holub et al. 2012)** [12] is another sophisticated method that embeds data in complex regions of a cover image using directional high-pass filters, while minimizing alterations in less textured areas. S-UNIWARD **(Holub et al. 2014)** [13] operates similarly to WOW but is designed for use in both spatial and non-spatial (e.g., frequency) domains. HILL (high-pass, low-pass, low-pass) **(Li et al. 2014)** [24] addresses the issue of distortion caused by embedding suitability in a single direction by using one high-pass filter along with two low-pass filters. By minimizing the power of the optimal detector (MiPOD), **Sedighi et al. 2015** [37] created a closed-form expression for the detector of content-adaptive least significant bit matching and embedded payloads. **Liao et al. 2020** [26] proposed two adaptive payload distribution strategies based on image texture features and distortion distribution in multiple image steganography, providing theoretical security analysis from the perspective of the steganalyst. The primary drawback of statistical methods is their relatively low bit encoding capacity (≤ 0.5 bpp).

Spatial domain

- **LSB (Least Significant Bit):** A simple steganographic technique that embeds secret data by replacing the least significant bits of the cover image pixels with bits of the secret message offers high capacity but low robustness against attacks.
- **PVD (Pixel Value Differencing):** Utilizes the difference in pixel values of ad-

jacent pixels to determine how much secret data can be embedded, allowing for more data in smoother areas and less in areas with higher contrast.

- **EMD (Exploiting Modification Direction):** Embeds secret data by slightly modifying the pixel values in a way that the direction of change represents the secret information, aiming to maintain visual quality.
- **RGB:** Involves hiding information in the RGB color channels of an image, where data can be embedded in one or more of the red, green, or blue components.
- **Huffman Encoding:** A lossless data compression algorithm that can be used in steganography to compress the secret message before embedding, reducing the amount of data to be hidden.
- **Mapping Based:** Involves mapping secret data to a set of patterns or structures within the cover image, often using a key for determining the mapping.
- **Code Based:** Uses error-correcting codes to embed secret data, providing robustness against image alterations that may occur during transmission or storage.
- **Spread Spectrum:** Hides data by spreading it across the frequency spectrum of the cover image, making it less susceptible to localized image modifications.
- **CRT (Chinese Remainder Theorem):** A mathematical approach used in steganography to embed secret messages by representing them as a set of remainders with respect to a chosen set of pairwise coprime integers.
- **Histogram Shifting:** Involves shifting the histogram of pixel values to create space for embedding secret data, with the goal of maintaining the statistical properties of the image.
- **Edge Detection Based:** Utilizes the edges in an image, which are areas with high pixel intensity variation, to embed data where it is less likely to be noticed.
- **Adaptive Steganography:** Techniques that adapt the embedding process based on the content of the cover image to improve imperceptibility and robustness.
- **Edge Adaptive Steganography:** Focuses on embedding information in the edges of an image where changes are less noticeable.
- **F5 Algorithm:** A steganographic algorithm that embeds data by modifying the coefficients of a JPEG image's discrete cosine transform (DCT).
- **Quick Response Codes (QR Codes):** Although not a steganography technique

per se, QR codes can be used to hide information in plain sight, as the data is encoded in a machine-readable optical label.

Frequency domain

- **Discrete Cosine Transform (DCT):** It modifies the DCT coefficients to embed data in the frequency domain, which is used in JPEG compression.
- **Discrete Fourier Transform (DFT):** Embeds information by modifying the magnitude or phase components of the image's Fourier transform.
- **Discrete Wavelet Transform (DWT):** Hides data within the wavelet coefficients obtained from the wavelet transformation of the image.
- **Singular Value Decomposition (SVD):** Embeds data by altering the singular values of the image matrices obtained through SVD.
- **Z-Transform:** Applies the Z-Transform to blocks of the image and embeds data into the transformed coefficients.
- **IWT (Integer Wavelet Transform):** A variant of the wavelet transform that uses integer calculations instead of floating-point. It's advantageous for steganography as it allows for exact reconstruction of the original image, which is crucial for lossless data embedding and extraction.
- **FFT (Fast Fourier Transform):** An efficient approach for calculating the inverse of the Discrete Fourier Transform (DFT). FFT can be used in steganography to convert an image into the frequency domain, where data can be inserted with the least amount of alteration to the image's perceived quality.
- **DCVT (Discrete Cosine and Vertex Transform):** While not as commonly referenced as DCT (Discrete Cosine Transform), DCVT may refer to methods that combine discrete transforms for embedding data. These methods leverage the strengths of different transforms to improve the steganographic process.
- **Entropy Thresholding:** A technique that uses entropy measures to determine suitable areas for embedding in the frequency domain.
- **Hybrid Techniques:** Combines spatial and frequency domain methods along with cryptography for enhanced security.

2.4.2 Deep Learning

CNN-Based

Convolutional Neural Networks (CNNs) are widely utilized in image steganography due to their capacity to learn hierarchical features from images. CNN-based steganography typically employs an encoder-decoder framework, where the encoder network embeds secret information into the cover image and the decoder network retrieves it from the stego image. The general implementation involves training the network to minimize the difference between the cover and stego images while ensuring accurate decoding of the secret information.

Baluja et al. 2017 [2] pioneered the use of deep neural networks to embed a full-size color image within another image of the same size, simultaneously training networks for the hiding (encoder) and revealing (decoder) processes designed to work as a pair. This was further extended in [3] to conceal multiple images and enhance hiding security by permuting the pixels of the secret image. **Rahim et al. 2018** [34] introduced a new loss function for joint end-to-end training of encoder-decoder networks. **Wu et al. 2018** [48] achieved a decoding rate of 98.2

Several methods have also explored hiding messages in physical photographs. **Wengrowski et al. 2019** [47] developed Light Field Messaging (LFM), a process for embedding, transmitting, and receiving hidden information in video displayed on a screen and captured by a handheld camera, aiming to minimize perceived visual artifacts while maximizing message recovery accuracy. This involves three networks for embedding, recovering, and transmitting, all using dense blocks with feature maps at different scales in a U-Net configuration. **Tancik et al. 2020** [39] introduced StegaStamp, a learned steganographic algorithm that robustly encodes and decodes arbitrary hyperlink bitstrings into photos, using a deep neural network with a U-Net style architecture robust to image perturbations from real printing and photography. **Jia et al. 2020** [16] proposed a U-Net based method to embed invisible hyperlinks into images, incorporating a distortion network between the encoder and decoder to maintain hidden messages resilient to cameras. **Jia et al. 2022** [17] presented a novel U-Net based architecture for invisible information hiding in display/print-camera scenarios, which includes hiding, locating, correcting, and recovering, with learned invisible markers and localized hidden codes.

Zhang et al. 2019 [55] proposed a novel CNN architecture named ISGAN, which hides a secret gray image within a color cover image and extracts the secret image on the receiver side. They enhanced invisibility by hiding the secret image only in

the Y channel of the cover image and developed a mixed-loss function more suited for steganography, resulting in more realistic stego images and better-revealed secret images.

GAN-based

Generative Adversarial Networks (GANs) are used in steganography to create stego images that are indistinguishable from cover images. A GAN-based steganography system comprises two competing networks: a generator, which creates stego images, and a discriminator, which attempts to distinguish between cover and stego images. The training process aims to reach a Nash equilibrium, where the generator produces stego images that the discriminator cannot distinguish from cover images.

Tang et al. 2017 [40] introduced an Automatic Steganographic Distortion Learning framework using GANs (ASDL-GAN) for spatial cover images. This framework consists of a steganographic generative sub-network and a steganalytic discriminative sub-network, where the implicit distortion function is directly related to undetectability against an evolving steganalyzer. Around the same time, **Hayes et al. 2017** [10] applied adversarial training techniques to learn a steganographic algorithm through unsupervised training. Their supervised training produced a robust steganalyzer but was limited to fixed-size images and suffered in quality when encoding beyond 0.4 bits per pixel.

Zhu et al. 2018 [59] improved on these approaches by using the same loss functions but modifying the model architecture to handle arbitrary-sized images, though it still struggled with higher relative payloads. GANs were used by **Zhang et al. 2019** [55] to reduce the divergence between the empirical probability distributions of stego pictures and natural images, hence improving the security of ISGAN. **Zhang et al. 2019** [54] presented a method for employing GANs to conceal arbitrary binary data in photos that same year. This technique avoided detection by steganalysis tools, produced state-of-the-art payloads of 4.4 bits per pixel, enhanced the perceptual quality of the images, and worked well on images from various datasets.

INN-based

Jing et al. 2021 [18] introduced HiNet, an innovative framework based on invertible neural networks (INNs) designed to address three major challenges in image hiding: large capacity, high invisibility, and hiding security. HiNet achieves large capacity through an inverse learning mechanism that simultaneously learns the image concealing and revealing processes. It ensures high invisibility by embedding secret in-

formation in the wavelet domain instead of the pixel domain and enhances hiding security by using a new low-frequency wavelet loss to ensure that secret information is hidden in high-frequency wavelet subbands.

Lu et al. 2021 [30] suggested a large-capacity steganography system called the Invertible Steganography Network (ISN). The recovery of concealed pictures and steganography are treated by ISN as two inverse issues involving image domain transformation. To handle the picture embedding and extraction tasks, the forward and backward propagation operations of a single invertible network are utilized.

In the steganographic realm, PIRNet is a novel method for privacy-preserving picture restoration that was introduced by Deng et al. in 2023 [7]. PIRNet hides a secret image inside a stego image using an Invertible Hiding (LIH) network based on wavelet Lifting. It then performs picture restoration within the steganographic domain using a Lifting-based Secure Restoration (LSR) network.

In addition to these methods, **Zhang et al. 2020** [53] introduced a novel universal deep hiding (UDH) meta-architecture. This approach disentangles the encoding of the secret image from the cover image, and upon independent analysis of the encoded message, they found that the success of deep steganography is due to a frequency discrepancy between the cover image and the encoded secret image. Despite being cover-agnostic, UDH achieves performance comparable to existing cover-dependent deep hiding (DDH) methods and supports a flexible number of images or channels for secret and cover.

Ghamizi et al. 2021 [9] introduced EAST, a multi-label targeted evasion attack-based steganography and watermarking method. Using this technique, data is encoded as the labels of the picture that the evasion attacks create.

Kishore et al. 2021 [22] introduced Fixed Neural Network Steganography (FNNS), a novel algorithm that leverages the sensitivity of neural networks to tiny perturbations. When compared to earlier state-of-the-art techniques, FNNS yields considerably lower error rates, reliably hides up to 3 bits per pixel (bpp) of secret information with a 0% error rate, and successfully evades existing statistical and neural steganalysis systems.

2.4.3 Generative

Generative steganography involves using generative models to synthesize images directly from secret messages without relying on cover images. This approach offers several advantages over traditional steganography methods that modify cover images. One key benefit is that it can evade detection by steganalysis methods since it does not alter existing images. Additionally, steganalysis methods trained on such images cannot detect hidden data. First presented in [58], the inventors of this concept indexed a collection of cover photos to distinct hash values that could be converted into secrets. Early generative steganography methods hid messages in simple images, such as textures or fingerprints. However, these methods often produced low-quality and unnatural images, making them susceptible to detection by third parties.

GAN-based

Generative steganography approaches have been developed to create high-quality and natural stego images, particularly utilizing GAN models. **Liu et al. 2017** [28] and **Zhang et al. 2022** [56] embed messages in the label embedding of conditional GANs. **Wang et al. 2018** [42], **Hu et al. 2018** [15], **Yu et al. 2021** [50], and **Wei et al. 2022** [43] trained new extractor models. **Shi et al. 2018** [38] introduced a novel strategy called Secure Steganography based on Generative Adversarial Networks (SSGAN), which generates suitable and secure covers for steganography. This strategy includes one generative network to evaluate the visual quality of the images and two discriminative networks to assess their suitability for information hiding.

In SEGAN by **Volkhonskiy et al. 2020** [41], the authors trained the system to convert the secret message into GAN latent noise, which generates appropriate container images. **Chen et al. 2022** [6] proposed a different computational framework using a deep neural network (DNN), specifically a SinGAN, a pyramid of GANs, to model the probability density of cover images and hide a secret image in a specific location within the learned distribution.

AutoEncoder-based

Liu et al. 2022 [29] introduced Image DisEntanglement Autoencoder for Steganography (IDEAS), a novel steganography without embedding (SWE) technique. **Bui et al. 2023** [5] proposed RoSteALS, a practical steganography method that uses frozen pretrained autoencoders to separate payload embedding from learning the distribution of cover images. RoSteALS can be adapted for innovative cover-less steganography applications, where the cover image is sampled from noise or conditioned on text

prompts via a denoising diffusion process.

INN-based

Zhou et al. 2022 [57] and **Wei et al. 2022** [44] proposed using invertible Flow models to achieve a high capacity for hidden messages. **Xu et al. 2022** [49] presented RIIS, a novel flow-based methodology for resilient invertible image steganography that models the distribution of the redundant high-frequency component based on the container picture through the use of a conditional normalizing flow.

Existing studies on generative steganography typically use GAN or Flow models to achieve high hiding capacity and strong anti-detection capabilities. However, GAN-based methods struggle to fully recover hidden data due to their lack of invertibility, while Flow-based methods often result in poor image quality because of the strict reversibility requirements in each module.

Diffusion models, which generate images through a stochastic iterative denoising process from Gaussian noise, offer high-quality image generation despite their high computational cost. To mitigate this, several studies have introduced deterministic sampling methods for diffusion models, which aim to eliminate the stochastic nature of these models while maintaining an invertible sampling process.

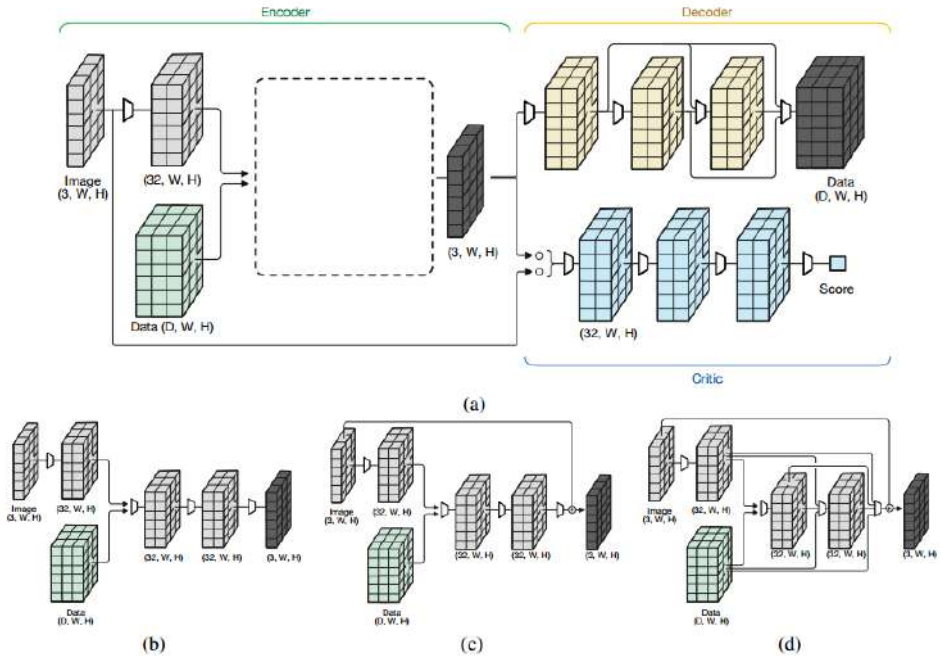
Kim et al. 2023 [21] proposed DiffusionStego, a generative steganography approach based on diffusion models, which outperforms other generative models in image generation. **Wei et al. 2023** [45] introduced "Generative Steganography Diffusion" (GSD) by developing an invertible diffusion model named "StegoDiffusion". **Yu et al. 2023** [52] proposed CRoSS (Controllable, Robust, and Secure Image Steganography), which offers significant advantages in controllability, robustness, and security over cover-based methods using diffusion models like Stable Diffusion, without requiring additional training.

Nguyen et al. 2023 [32] introduced Stable Messenger, which uses a new latent-aware encoding technique leveraging pretrained Stable Diffusion for advanced steganographic image generation. This method achieves a better trade-off between image quality and message recovery. They also proposed two new metrics: "message accuracy," which evaluates the entirety of decoded messages for a more comprehensive assessment, and "Log-Sum-Exponential (LSE) loss," an adaptive universal loss designed to enhance message accuracy.

2.5 String embedding

2.5.1 SteganoGAN *[MIT EECS, 2019]*

In this paper [54], the authors introduce a novel strategy that addresses the task of steganography through adversarial training, achieving a relative payload of 4.4 bits per pixel, which is 10x greater than competing deep learning-based systems with comparable ratios of peak signal to noise. To compare deep learning-based steganography algorithms to more conventional methods, they provide a new criteria for assessing their capacity. The authors assess their method by calculating how well it can avoid various statistical as well as deep-learning based steganalysis instruments that are intended to identify if an image is steganographic or not. Most conventional steganalysis methods only achieve a detection auROC of < 0.6, even when > 4 bits per pixel are encoded into the image. Using data produced by their approach, they train a cutting-edge model for automatic steganalysis proposed by Ye et al. (2017). Even if they mandate that their model generate steganographic images with a detection rate of no more than 0.8 auROC, they discover that their model is still capable of concealing up to two bits per pixel. To assess deep learning-based steganography methods, they are making available STEGANOGAN, a fully-maintained open-source library that comes with datasets and pre-trained models.



(a) The model architecture with the Encoder, Decoder, and Critic. The blank rectangle representing the Encoder can be any of the following: (b) Basic encoder, (c) Residual encoder and (d) Dense encoder. The trapezoids represent convolutional blocks, two or more arrows merging represent concatenation operations, and the curly bracket represents a batching operation.

Figure 2.1: Architecture overview of Steganogan

As illustrated in Figure 2.1, the architecture consists of three modules: (1) An Encoder creates a steganographic image from a cover image and a data tensor, or message; (2) A Decoder uses the steganographic image to try and recover the data tensor; and (3) A Critic assesses the quality of the steganographic and cover images. The encoder further has three variants, 1) Basic, 2) Residual and 3) Dense. We can see the architectural difference of these 3 variants in Figure 2.1.

The authors present an adversarial Critic to offer input on the encoder's performance and produce more lifelike images. Three convolutional blocks make up the critic network, which is then followed by a convolutional layer with one output channel. The adaptive mean pooling is utilized to generate the scalar score by applying it to the convolutional layer's output.

The model was trained using DIV2K and COCO datasets. During experimentation, the authors notice that all of their model's variations perform better on the COCO dataset than the DIV2K dataset. This can be explained by variations in the kinds of content that were captured on camera for the two datasets. While photos from the COCO dataset tend to be more congested and feature several items, images from the DIV2K dataset typically contain open scenery, giving their model more surfaces and textures to successfully embed data on. Furthermore, the authors observe that the residual variant, which displays comparable image quality but a lower relative payload, trails closely behind their dense variant, which performs best in terms of both relative payload and image quality. When compared to the dense variation, the basic variant performs the lowest on all parameters, with relative payloads and image quality scores that are 15–25% worse.

Figure 2.2: Results after testing on a random image from the COCO Dataset. The left image is the cover image and the right one is the stego image

The code was made public and we were able to reproduce the result. In the Figure 2.3, we present to you the result of our experiments. We ran the code and got the provided

outputs. The Figure 2.3(a) shows the input image. We encoded a string using the Basic variant and got the output Figure 2.3(b). We again trained the model using the Dense variant, and then got the output Figure 2.3(c). As the authors said, the basic variant gave the worst performance, we also found that the stego image from basic had heavy color distortions. Similarly, we got best results from the Dense variant, which has negligible differences with the cover image.

2.5.2 StegaStamp *[CVPR, 2020]*

In this paper [39], the authors introduce StegaStamp, a novel approach to embedding imperceptible digital data, specifically hyperlinks, into printed and digitally displayed photos. In order to achieve nearly perceptual invisibility, the technique uses a deep neural network to learn a steganographic algorithm that guarantees reliable encoding and decoding of linked bitstrings. Under a variety of real-world conditions, such as changes in lighting, shadows, perspective, occlusion, and viewing distance, the system exhibits real-time decoding capabilities. The first end-to-end trained deep pipeline for delivering robust decoding even in physical transmission scenarios—like actual printing or display—is introduced in the study as StegaStamp. The addition of differentiable pixelwise and spatial image corruptions contributes to approximating distortions encountered in physical transmission, resulting in robust retrieval of 95% of encoded bits with excellent perceptual image quality. The prototype's capability to uniquely encode hyperlinks for a significantly larger number of images than currently available on the internet is highlighted, emphasizing its potential impact on data transmission in diverse image scenarios.

(a) Input image **(b)** Stego image using Basic

(c) Stego image using Dense

Figure 2.3: Our results from training SteganoGAN

Figure 2.4: An overview of the system in the context of a typical usage flow

Figure 2.4 illustrates an overview of the system's typical usage flow. The process begins with an image and a desired hyperlink as inputs. Firstly, the hyperlink undergoes conversion into a unique bit string, akin to the approach used by URL-shortening services such as tinyurl.com. Next, the bit string gets concealed within the image using the StegaStamp encoder. This results in an encoded image that appears identical to the original. Subsequently, the encoded image is either printed on paper or displayed on a screen, making it accessible in the physical world. Users then capture a photo of the printed image as the fourth step. The system employs an image detector in the fifth step to locate and crop the images within the photo. Finally, utilizing the StegaStamp decoder, the system extracts the bit string from each image and proceeds to follow the hyperlink, thereby accessing the online content associated with the image.

The proposed StegaStamp method comprises three main components: 1) Encoder utilizes a U-Net style architecture to embed a 100-bit binary message into a three-channel 400×400 pixel input image. It produces a three-channel RGB residual image aimed at minimizing perceptual differences. 2) Decoder employs a spatial transformer network to enhance robustness against minor perspective changes. It retrieves the hidden message from the encoded image using convolutional and dense layers with a sigmoid activation function. 3) In practical scenarios, a BiSeNet-based detector is employed to detect and correct StegaStamps within a wide field-of-view image before decoding. This ensures efficient processing, particularly for larger images.

The encoding network and decoding network are jointly trained using images from the MIRFLICKR dataset at a resolution of 400×400 pixels, along with random binary messages. For training the BiSeNet detector, randomly transformed StegaStamps from the DIV2K dataset are used.

During the training process, the model approximates the effects of a physical display-imaging pipeline to ensure robustness for real-world applications. Specifically, the output of the encoding network undergoes random transformations such as Perspective Warp, Motion and Defocus Blur, Color Manipulation, Noise addition, and JPEG

(a) Original Image **(b)** StegaStamp **(c)** Residual

Figure 2.5: Images that have been encoded. To create the encoded StegaStamp, the encoder network computes the residual and adds it back to the original image. These samples are resistant to picture disturbances caused by the printing and imaging procedures and have messages encoded with 100 bits.

Compression. These transformations are applied before passing the image through the decoding network. This approach helps the model learn to handle distortions that may occur during the display and imaging process, thereby improving its practical usability.

The encoder/decoder pipeline includes a critic network for perceptual loss. Wasserstein loss guides critic training, interleaved with encoder/decoder training. L2 residual regularization (L_R), LPIPS perceptual loss (L_P), and critic loss (LC) minimize perceptual distortion. Cross-entropy loss (L_M) handles the message and is used to supervise the decoder. Training loss (L) is a weighted sum ($L = \lambda_R L_R + \lambda_P L_P + \lambda_C L_C + \lambda_M L_M$). Adjustments involve gradually increasing image loss weights ($\lambda_{R,P,C}$), perturbation strengths, and mitigating distracting patterns at image edges.

The system is tested in both real-world and simulated scenarios (Figure 2.6), achieving a mean bit-accuracy of 98.7% across diverse settings. Real-world testing on cellphone camera videos demonstrates robust decoding, even when StegaStamps are partially obscured. Highly robust decoding is confirmed across 18 combinations of displays/printers and cameras, yielding a mean accuracy of 98.7% over 1890 images. Training with both pixelwise and spatial perturbations proves significantly more effective than models with no or single-type perturbations. A message length of 100 bits strikes a balance between recovery accuracy and perceptual similarity, enabling encoding of 56 error-corrected bits, as shown in Table 2.2.

In their quest to develop models resilient to real-world conditions, the authors explore various methods. These methods involve subjecting the model to differentiable image perturbations during training, mimicking the distortions encountered in the physical display and imaging processes. Notably, prior studies have employed similar

Figure 2.6: Examples of the system in use in natural settings. Message recovery accuracies are shown together with an outline of the StegaStamps that were found and deciphered.

Table 2.2: The average image quality across 500 photographs was used to train models with varying message lengths. A higher PSNR and SSIM are preferable. Lower is better in LPIPS, a learnt perceptual similarity metric.

Metric	**Message length**			
	50	100	150	200
PSNR	29.88	28.50	26.47	21.79
SSIM	0.930	0.905	0.876	0.793
LPIPS	0.100	0.101	0.128	0.184

techniques to enhance the robustness of classification networks against adversarial attacks, a concept termed "Expectation over Transformation."

Drawing from previous works such as HiDDeN and Deep ChArUco, they integrate both spatial and non-spatial perturbations to enhance the resilience of their encoder-

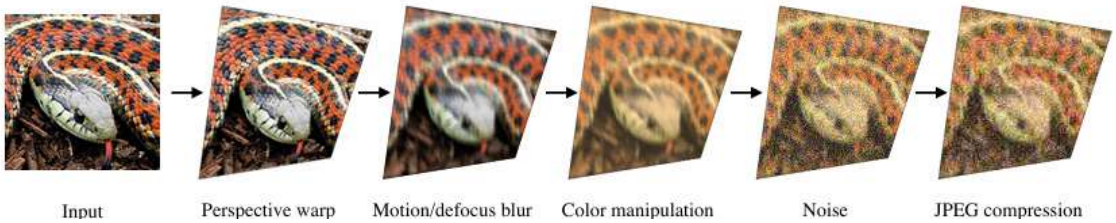


Figure 2.7: Pipeline for image perturbation. To make the model reliable for usage in real-world scenarios, the authors simulate the impacts of a physical display-imaging pipeline during training. Before sending the picture via the decoding network, they take the encoding network's output and apply the arbitrary perturbations displayed here.

decoder system, particularly in transmitting concealed messages through physical display and imaging setups.

The perturbation pipeline encompasses several components:

- **Perspective Warp**: Simulating the effects of a camera's misalignment by generating random homographies, thereby warping the image perspective.
- **Motion and Defocus Blur**: Mimicking blur induced by camera motion or focus inaccuracies by applying random blur kernels.
- **Color Manipulation**: Introducing variations akin to imaging noise and adjustments made by cameras, such as Gaussian noise and color transformations.
- **Noise**: Incorporating noise typical of camera systems, modeled by Gaussian noise.
- **JPEG Compression**: Replicating the lossy compression encountered in storing images, employing a differentiable approximation to handle the non-differentiable quantization step.

By subjecting their model to these diverse perturbations during training, the authors aim to fortify its ability to operate effectively in real-world scenarios, where such distortions are commonplace. This amalgamation of techniques from various studies underscores their commitment to developing robust systems capable of navigating the challenges posed by practical application environments.

We were able to reproduce the result with the pretrained model they have given. While successful, the system has perceptibility in low-frequency regions, and custom detection architecture optimization is identified as a potential improvement for real-world performance.

2.5.3 IDEAS *[CVPR, 2022]*

The paper [29] introduces Image DisEntanglement Autoencoder for Steganography (IDEAS), a novel Steganography Without Embedding (SWE) technique. Unlike conventional steganography, IDEAS transforms the secret message into a synthesized image, making it inherently resistant to steganalysis attacks. By disentangling images into structure and texture representations, IDEAS leverages the stability of structure for improved message extraction and enhances security by randomizing texture representations. The approach includes an adaptive mapping mechanism to different extraction levels for diverse image synthesis, demonstrating superior performance in

terms of security, reliable message extraction, and flexibility for different extraction levels compared to existing SWE methods.

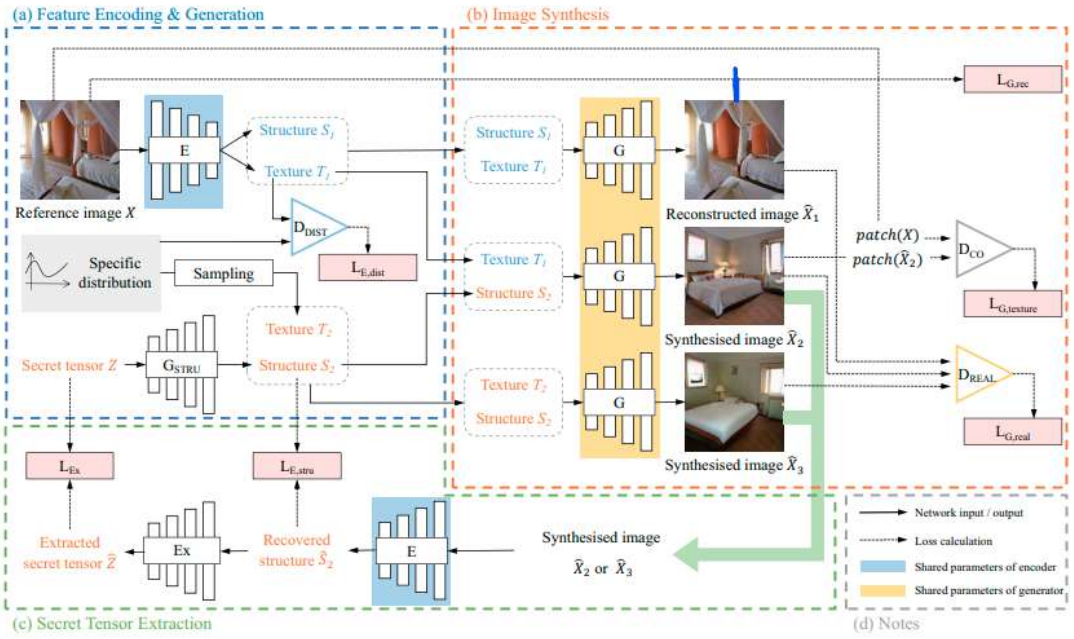


Figure 2.8: Training flowchart of IDEAS network

This paper presents the Image DisEntanglement Autoencoder for Steganography (IDEAS) as a remedy for the limitations seen in current synthesis-based Steganography Without Embedding (SWE) techniques. Figure 2.8 shows the training flowchart of the IDEAS network. IDEAS utilizes an adversarial autoencoder, including an encoder for the disentanglement of images into distinct structure tensor and texture vector representations. Then, a decoder uses the texture vector sampled from a uniform distribution and the structure tensor obtained from the secret tensor (i.e., the encoded secret message) to produce a realistic image for data concealment. In order to guarantee both excellent synthesis and low extraction error of the secret message, training entails simultaneously optimizing numerous loss terms and incorporating a variety of synthesis approaches. Here, StyleGAN2 [20] is used as the underlying architecture for the generator to achieve high-quality image synthesis. The suggested method seeks to address challenges associated with realism, synthesis diversity, and message extraction errors observed in existing synthesis-based SWE approaches.

In Figure 2.9 we can see to unveil the concealed message within XM, the recipient utilizes E and Ex for consecutive retrieval of $\hat{S}_M$ and $\hat{Z}_M$. Subsequently, the inverse mapping function

$$m = \lfloor (z+1) \times 2\sigma^{-1} \rfloor \tag{2.6}$$

is employed to extract the secret message $\hat{M}$. In contrast to existing SWE techniques,

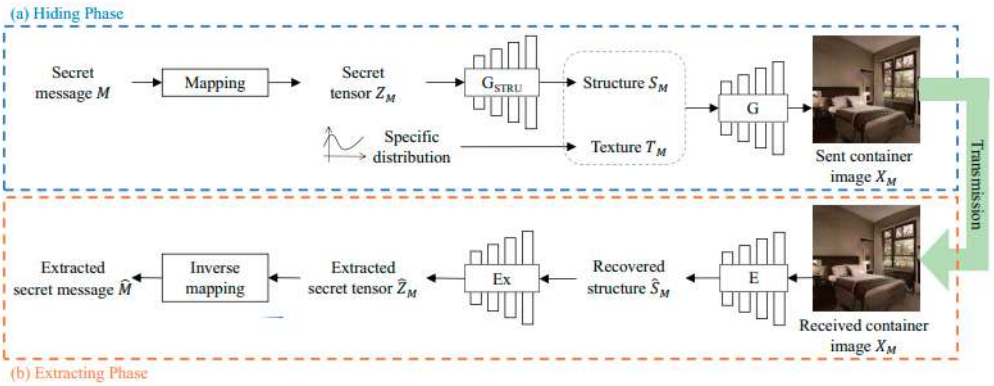


Figure 2.9: Diagram showing the flow of ideas for covert communication, divided into two sections: the concealment phase and the extraction phase

this method represents the secret message through the structure of a generated image, enhancing extraction accuracy due to structural stability. Furthermore, the container image synthesized by the StyleGAN2 generator G exhibits high fidelity, contributing to the heightened imperceptibility of secret message transmission.

To showcase the effectiveness of IDEAS, this paper conducted a comparative analysis against four contemporary synthesis-based Steganography Without Embedding (SWE) methods: DCGAN-Steg [15], SAGAN-Steg [50], SSteGAN [42], and WGAN-Steg [25]. IDEAS and baseline models undergo training on subsets extracted from publicly available datasets, specifically Bedroom and Church images from LSUN [51], and face images from FFHQ [19]. Each subset comprises 70,000 randomly selected images normalized to 256×256 pixels. IDEAS is trained with hyperparameters $N = \{1, 2\}$ and $\lambda = \{2, 5, 10\}$, with λ set to 10 for performance evaluation; other λ settings are discussed in Subsection 4.6. Evaluation metrics include the area under the curve (AUC) of the receiver operating characteristic (ROC) curves and the Frechet Inception Distance (FID) [11] to assess undetectability by steganalysis tools and subjective visual perception. Additionally, the secret message extraction accuracy of IDEAS is compared under various hidden capacities with benchmark SWE methods. All computations are conducted on an RTX 2080Ti, with IDEAS, comprising a total of 64.5M parameters, taking 12.0 ms for image synthesis and 9.4 ms for message extraction.

Table 2.3: Comparison of Steganalysis Results for IDEAS ($N = 1$, $N = 2$), DCGAN-Steg, SAGAN-Steg, SSteGAN, and WGAN-Steg (% Distortion)

Method	IDEAS ($N = 1$)			IDEAS ($N = 2$)			DCGAN-Steg	SAGAN-Steg	SSteGAN	WGAN-Steg
Distortion	0	25	50	0	25	50				
StegExpose	0.480	0.502	0.542	0.456	0.471	0.484	0.586	0.578	0.417	0.594
XuNet	0.404	0.398	0.371	0.413	0.407	0.403	0.568	0.500	0.491	0.542
YeNet	0.521	0.512	0.520	0.533	0.528	0.535	0.573	0.569	0.519	0.548

Figure 2.10: Examples of synthesised container pictures on LSUN Bedrooms (left), LSUN Churches (middle), and FFHQ (right) from DCGAN-Steg, SAGAN-Steg, SSteGAN, WGAN-Steg, and IDEAS. (A) a comparison of the image fidelity synthesized from several covert communications. (b) a comparison of the variety of images created using the same secret message.

Figure 2.10 shows the qualitative results of the author.

We trained the model using FFHQ dataset and were not able to reproduce quite satisfactory results. As we can see that there are noticeable structural differences in Fig-

Figure 2.11: Our Results from Training IDEAS

ure 2.11 and we did not get quite satisfactory results by reproducing this paper IDEAS.

While IDEAS exhibits exceptional performance, its hidden capacity is relatively smaller in comparison to certain techniques, such as full-image-to-image hiding methods. As we were not successful in this training phase we explored other options for delving into more details about steganography.

2.6 Image embedding

2.6.1 HiNet *[ICCV, 2021]*

A novel framework called HiNet, which is built on invertible neural networks (INNs), was proposed to handle the three challenges of image hiding: security, invisibility, and capacity. With an improvement of over 10 dB PSNR in secret picture recovery on ImageNet, COCO, and DIV2K datasets, HiNet performs noticeably better than state-of-the-art image hiding methods. HiNet uses inverted learning mechanism in which disclosing and hiding operate in essentially opposite ways. HiNet also explains why hiding a secret image in the frequency domain is better than hiding it in the pixel domain. This paper proposed to use a low-frequency wavelet loss to control the diffusion of sensitive data over a wide range of frequencies, resulting in a significant improvement in the security of concealment.

Conventional steganographic techniques are unable to meet the requirements of big capacity in picture hiding tasks because they can only conceal a limited quantity of information [4], [8], [14], [23], [31], [35]. All these conventional methods introduced two separate sub-networks: one for the concealing and the other for the recovery purpose. For image hiding, all these methods use two sub-networks: a revealing network that recovers the secret image (x_{rec}) from the stego image, and a concealing network that embeds a secret image (x_{secret}) into a cover image to create a stego image (x_{stego}). The two networks that are hiding and revealing have two sets of parameters that are connected by a straightforward concatenation. Color distortion and texture-copying artifacts could result from this poor relationship. Furthermore, these techniques frequently ignore security concerns, making the concealed secret information easily detectable.

This is where the novelty of this paper HiNet comes in. Here, secret image revealing is modeled as the invertible network architecture's opposite of image concealing, meaning the network only needs to be trained once to acquire all network parameters for both revealing and concealing. Due to its excellent performance INN has been used in a lot of image processing tasks but HiNet is the first to use INN for the image hiding purpose.

Figure 2.12 shows the network architecture of HiNet. Here in the concealing process we have 16 consecutive concealing blocks. For the very first block the secret image (x_{secret}) and the cover image (x_{cover}) after being transformed into the frequency domain by DWT are served as inputs. After applying the Discrete Wavelet Transform (DWT), the feature map with dimensions (B, C, H, W) is transformed into a new feature map

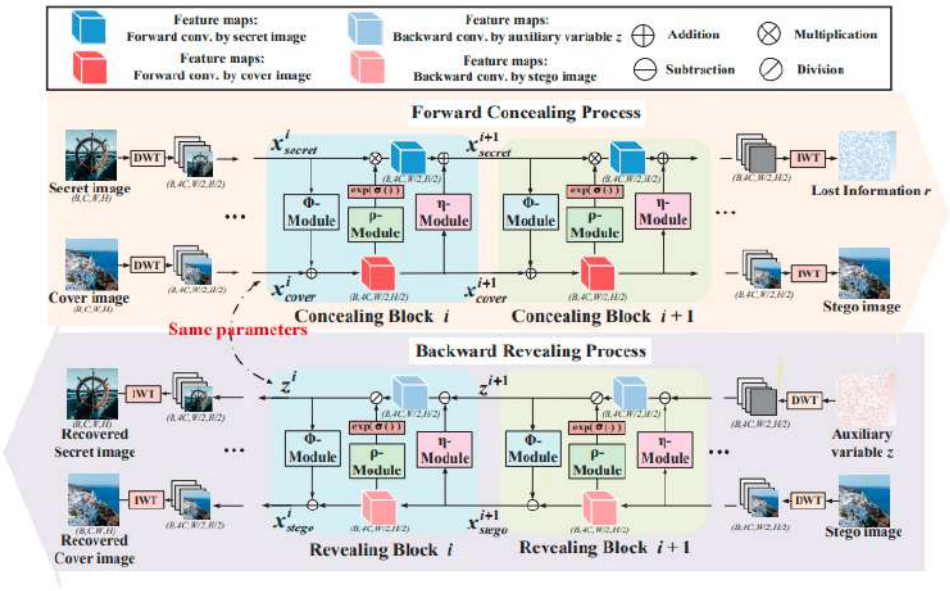


Figure 2.12: The architecture of HiNet

with dimensions (B, 4C, H/2, W/2). Here, B represents the batch size, H is the height, W is the width, and C is the channel number. The use of DWT results in a reduction in computational cost. Then after the 16th block after IWT we get the lost information r and the stego image. The revealing process is the inverse of the concealing process which takes the stego image (x_{stego}) and auxiliary variable z and we get the retrieved secret (x_{rec}) as the output.

HiNet tries to minimize three kind of losses : concealing loss, recovery loss and low-frequency wavelet loss. The concealing process incorporates a concealing loss to ensure effective concealment, while the revealing process involves revealing losses to guarantee successful recovery. Additionally, a novel low-frequency wavelet loss is introduced to bolster the security of the hiding mechanism.

HiNet is tested on DIV2K [1] , ImageNet [36], and COCO [27] datasets after being trained on the DIV2K [1] dataset. To ensure uniform resolution, testing photographs are cropped in the center. The model uses a training patch size of 256x256, 16 concealing and revealing blocks ($M = 16$), and 80,000 iterations in total. The values of λ_c, λ_r, and λ_f are 10.0, 1.0, and 10.0, correspondingly. Half of the 16 mini-batch patches are utilized as cover patches, and the remaining half are hidden patches. Every 10,000 iterations, the Adam optimizer with standard parameters and a starting learning rate of $1 \times 10^{-4.5}$ is used.

The quality of cover/stego and secret/recovery pairings is evaluated using four metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Structural Sim-

Table 2.4: Benchmark comparisons on different datasets.

Methods	DIV2K				COCO				ImageNet			
	PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE
4bit-LSB	33.19	0.9453	6.90	7.95	33.79	0.9479	7.31	9.12	33.68	0.9401	6.46	8.48
HiDDeN [59]	35.21	0.9691	6.98	6.82	36.71	0.9876	6.58	8.73	34.79	0.9380	6.12	7.33
Weng et al. [46]	39.75	0.9765	3.24	4.85	38.89	0.9762	3.99	5.91	37.62	0.9588	4.70	5.25
Baluja [2]	36.77	0.9645	3.79	5.02	36.38	0.9563	5.98	7.43	36.59	0.9520	5.61	5.41
HiNet	48.99	0.9971	1.33	1.94	46.52	0.9961	1.87	2.92	44.60	0.9928	2.52	3.62

ilarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR). Better image quality is indicated by higher PSNR and SSIM values as well as lower RMSE and MAE values. In comparison to the other four methods, HiNet not only demonstrates superior re-

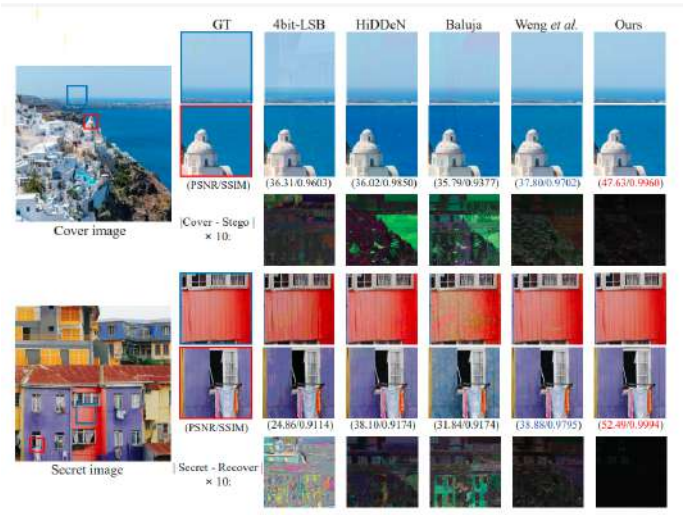


Figure 2.13: HiNet and other comparison methods, such as 4bit-LSB, HiDDeN [59], Baluja [3], and Weng et al. [46], were used to visually compare stego and recovery images. Larger stego images are shown in the top three rows, and larger recovery images obtained using various techniques are shown in the bottom three rows.

covery accuracy but also exhibits high color fidelity without text-copying artifacts in both stego and recovery images.

We also trained the model using the DIV2K [1] dataset and after testing using the DIV2K [1] dataset we were able to obtain almost similar results to that of the original results published by the author. Figure 2.14 shows the results that we were able to

(a) Cover **(b)** Secret **(c)** Stego **(d)** Recovered

Figure 2.14: Our Results from Training StegaStamp

reproduce after training the model. As we can see the results that we obtained are almost similar to that of the original ones. And as we can observe even after concealing the secret image into the cover image there is no such residual left in the stego image and no visible visual artifacts which proves the great hiding capacity of HiNet. Similarly the retrieved secret is almost similar to that of the original secret conceled which proves that HiNet has good revealing capacity too.

2.6.2 CRoSS *[NeurIPS, 2023]*

In this paper [52], the authors point out the drawbacks of the current image steganography techniques and suggest a single, overarching objective: security, controllability, and robustness. They further show that, by adopting a diffusion-based invertible image translation technique, the diffusion model may easily integrate with image steganography to accomplish these objectives without the need for extra training. Controllable, Robust, and Secure Image Steganography (CRoSS) is the new framework for image steganography that the authors propose. To the best of their knowledge, this is the first effort to improve performance in the field of image steganography by applying the diffusion model. The authors enhanced the controllability and diversity of CRoSS by proposing variants employing prompts, LoRAs, and ControlNets, capitalizing on the growth of the rapidly expanding Stable Diffusion community. They carried out extensive tests with a focus on the three goals of security, controllability, and robustness, showcasing the benefits of CRoSS over current approaches.

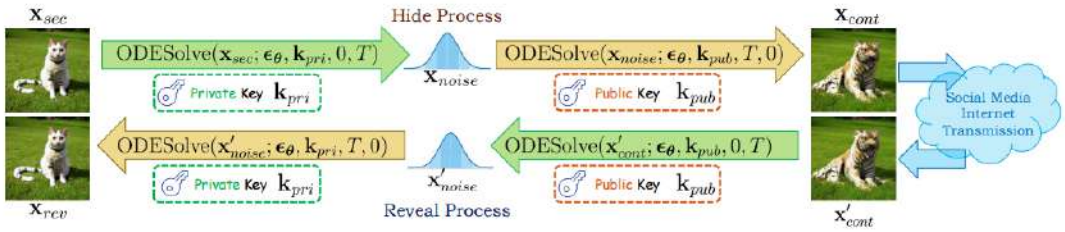


Figure 2.15: Coverless image steganography framework CRoSSS

Instead of embedding the secret image into a cover image, the secret image is translated to a stego/container image using a conditional diffusion model, implementing the DDIM technique as sampling strategy for invertibility. Diffusion models, being essentially Gaussian denoisers, naturally exhibit robustness to noise and perturbations, making them less susceptible to transmission issues. The complete diffusion model process involves two stages. The forward phase (i) generates a prompt or private key from the secret image, (ii) generates noise from secret image and the private key, (iii) adds noise to a clean image, and then (iv) generates the public key from the

user-provided prompt, finally (v) creates container image from the noise and the public key. The backward sampling phase denoises the container image step by step by reversing the procedure. Stable diffusion, along with the incorporation of LoRANets and ControlNets, contributes to better controllability and diversity.

Figure 2.16: Visual outcomes of the suggested CRoSS with various prompt settings. The disclosed images exhibit strong semantic consistency with the hidden images, and the container images appear lifelike.

The authors gather a benchmark consisting of 260 photographs from Google search engines and publicly accessible databases in order to conduct a quantitative and qualitative examination of their approach. They then create prompt keys that are specifically designed for the coverless image steganography, which they have named Stego260. The dataset is divided into three classes: namely people, animals, and common items (furniture, food, plants, architecture, etc.). In order to generate prompt keys, they first construct private keys using OpenAI's BLIP, then they make semantic alterations and generate public keys in batches using ChatGPT or artificial adjustment.

The experiments conducted by the authors involved different types of degradation, such as Gaussian noise, JPEG compression, and JPEG enhancer with varying quality factors. In all cases, the recovered secret image outperformed HiNet. In particular, they use WeChat's pipeline to send and receive container images in order to accomplish network transmission. They use a smartphone to take pictures of the container

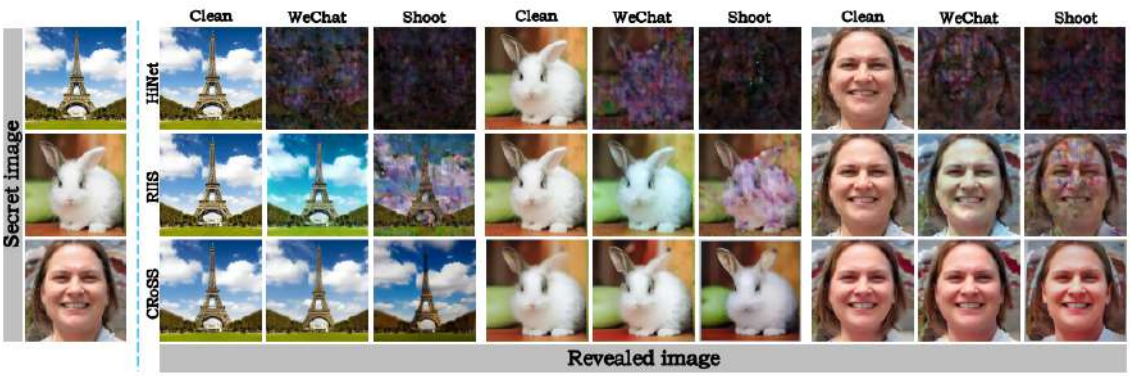


Figure 2.17: Visual comparisons between CRoSS and alternative techniques under two real-world degradations: "Shoot" and "WeChat".

Table 2.5: The PSNR (dB) results of the proposed CRoSS and other methods across various degradation levels show that CRoSS consistently achieves higher data fidelity in most scenarios.

Method	Clean	Gaussian Noise			Gaussian denoiser			JPEG Compression			JPEG Enhancer		
		$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$Q = 20$	$Q = 40$	$Q = 80$	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$Q = 20$	$Q = 40$	$Q = 80$
Baluja [2]	34.24	10.30	7.54	6.92	7.97	6.10	5.49	6.59	8.33	11.92	5.21	6.98	9.88
ISN [30]	41.83	12.75	10.98	9.93	11.94	9.44	6.65	7.15	9.69	13.44	5.88	8.08	11.63
HiNet [18]	42.98	12.91	11.54	10.23	11.87	9.32	6.87	7.03	9.78	13.23	5.59	8.21	11.88
RIIS [49]	43.78	26.03	18.89	15.85	20.89	15.97	13.92	22.03	25.41	27.02	13.88	16.74	20.13
CRoSS [52]	23.79	21.89	20.19	18.77	21.39	21.24	21.02	21.74	22.74	23.51	20.60	21.22	21.19

images on the screen at the same time, and then they just crop and distort them. Evidently, as Figure 2.17 illustrates, when subjected to these two incredibly complex degradations, all other methods have either completely failed or exhibit severe color distortion; however, their method is still able to reveal the approximate content of the secret images while maintaining good semantic consistency, demonstrating its superiority. From the Figure 2.17, it is evident that cross can reconstruct the content of secret images, whereas other approaches either fail miserably or show severe colour distortion.

Despite these achievements, there are still some limitations to CRoSS. Pixel-wise objective fidelity metrics, like PSNR, reveal a gap compared to cover-based methods because of the trade-off between invertibility and editing capability in diffusion-based zero-shot image translation.

2.7 Image Denoising

2.7.1 WINNet *[IEEE, 2022]*

In order to combine the benefits of learning-based and model-based techniques for image denoising, this research suggests a wavelet-inspired invertible network (WINNet). WINNet consists of a sparsity-driven denoising network, a noise estimation network, and lifting-inspired invertible neural networks (LINNs). For noise reduction, LINNs offer a non-linear redundant transform, whereas the denoising network makes use of sparse coding. Based on the amount of noise, the noise estimation network modifies the thresholds. High interpretability, robust generalization to various noise levels, and competitive performance in picture denoising and deblurring tasks are guaranteed by the architecture of WINNet.

- Proposes an invertible thresholding network for image denoising, inspired by wavelet-based methods, leading to high interpretability. LINNs create a non-linear redundant transform with perfect reconstruction, and the denoising network uses basis pursuit denoising.
- Achieves blind image denoising with a model-inspired noise estimation network that adapts soft-thresholds to the estimated noise level, providing strong generalization to unseen noise levels.
- Offers high interpretability and strong generalization, achieving competitive performance in non-blind/blind image denoising and image deblurring with simplicity and a small number of parameters.

The architecture of the WINNet is shown beolow. As shown in Figure 2.18 the LINN modules are used. The forward and inverse transforms separately are shown below. WINNet is trained and tested with the following settings.

- **Training Data:** 400 images from the BSD dataset, size 180×180.
- **Patch Sizes:** 40×40 for non-blind denoising, 50×50 for blind denoising.
- **Noise Levels:** Single noise levels $\sigma_N = 15, 25, 50$. For blind denoising, σ_N drawn from $[0, 55]$.
- **Regularization Parameters:** $\lambda_1 = 0.1$ (spectral norm loss), $\lambda_2 = 10$ (orthogonal loss), evaluated every 10 iterations.
- **Initialization:** Weights initialized using Kaiming method.
- **Optimizer:** Adam with initial learning rate $\text{lr} = 1 \times 10^{-3}$ and $\beta = (0.9, 0.999)$.

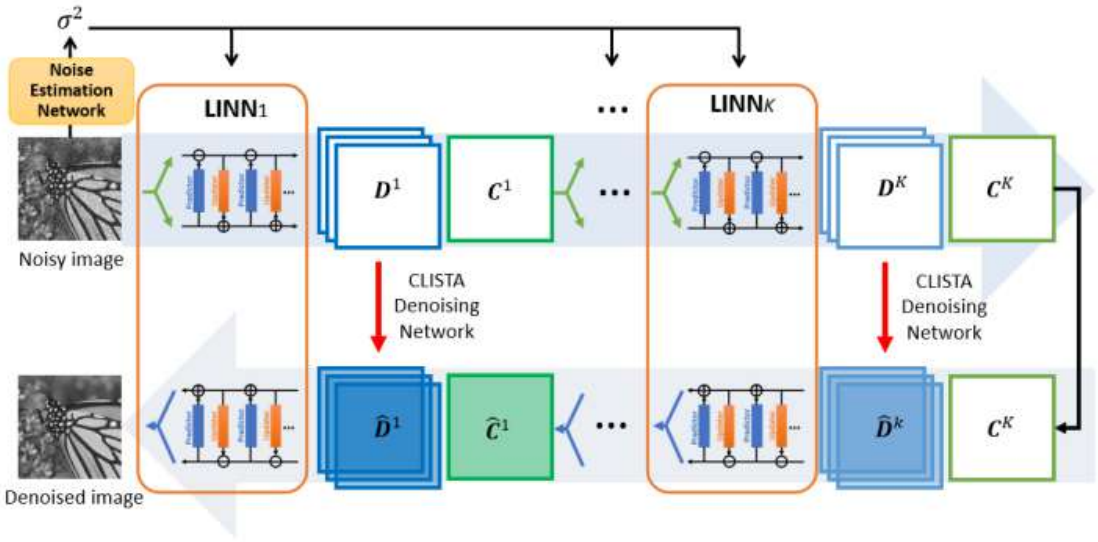
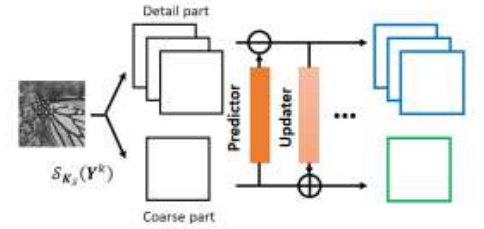
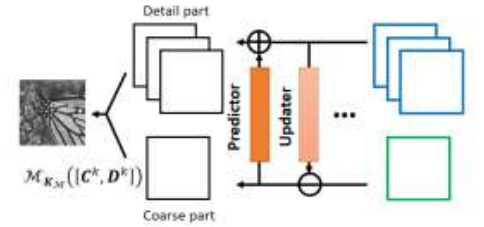


Figure 2.18: Overview of the proposed wavelet-inspired invertible network (WINNet)



(a) The forward transform of a LINN with splitting operator $\mathcal{S}_{\boldsymbol{K}_{\mathcal{S}}}(\boldsymbol{Y}^k)$.



(b) The inverse transform of a LINN with merging operator $\mathcal{M}_{\boldsymbol{K}_{\mathcal{M}}}([\boldsymbol{C}^k, \boldsymbol{D}^k])$.

Figure 2.19: The forward and inverse transforms of a LINN

- **Training Schedule:** 50 epochs, learning rate decays to 1×10^{-4} at the 30th epoch. Batch size $N = 32$.
- **Testing Data:** 12 images from Set12, 68 images from BSD68.
- **Evaluation Metric:** PSNR.

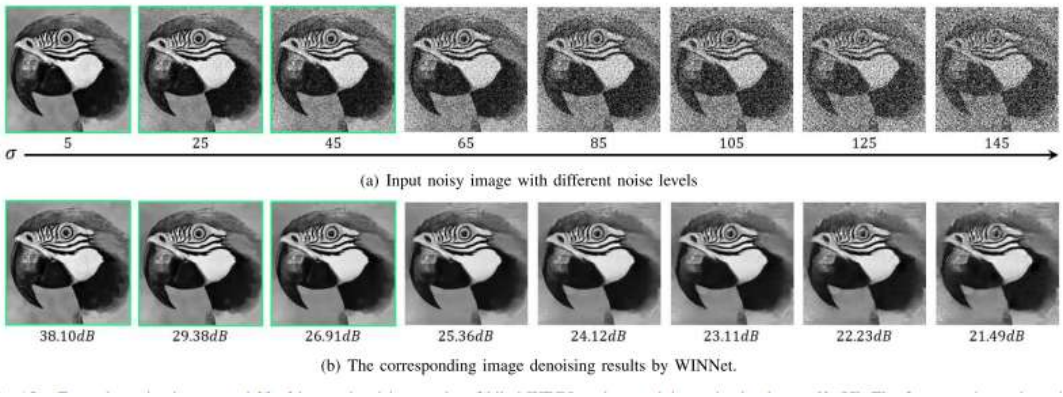


Figure 2.20: Utilizing blind WINNet and trained with noise levels $\sigma \in [0, 55]$, sample noisy images and the results of their denoising are presented. The second row shows the corresponding denoised results, whereas the first row shows noisy photos with different amounts of noise. Pictures that fall under the training noise levels have a green border around them.

kernel	1 	2 	3 	4 	5 	6 	7 	8 
EPLL [22]	31.47	31.00	31.23	29.80	32.47	32.28	31.12	30.53
IRCNN [6]	32.26	31.65	30.87	31.76	31.82	31.99	31.10	31.07
DRUNet [31]	33.26	32.88	32.76	32.71	33.72	33.88	33.04	32.70
Proposed	32.37	32.04	32.03	31.52	32.47	33.17	32.02	31.87

Figure 2.21: The Average PSNR (dB) of Various Image Deblurring Methods Evaluated on Set12 with a Noise Level of 2.55 (1%)

Dataset	Methods	Model Size	$\sigma = 5$	$\sigma = 25$	$\sigma = 45$	$\sigma = 65$	$\sigma = 85$	$\sigma = 105$	$\sigma = 125$	$\sigma = 145$
BSD68	DnCNN-B [25]	0.66M	37.75	**29.15**	26.62	23.00	16.07	13.19	11.68	10.79
	BUIFD [29]	1.19M	37.41	28.76	25.61	23.07	18.81	15.98	14.45	13.52
	BF-CNN [28]	0.66M	37.73	29.11	26.58	25.12	24.10	23.33	22.70	22.18
	WINNet (1-scale)	0.18M	**37.82**	29.13	**26.66**	**25.23**	**24.23**	**23.46**	**22.81**	**22.23**
Set12	DnCNN-B [25]	0.66M	37.88	**30.38**	27.68	23.52	15.95	13.18	11.78	10.92
	BUIFD [29]	1.19M	37.34	30.18	27.01	24.27	19.41	16.28	14.66	13.73
	BF-CNN [28]	0.66M	37.81	30.33	27.58	25.83	24.54	23.55	22.74	22.07
	WINNet (1-scale)	0.18M	**38.22**	30.33	**27.72**	**26.03**	**24.77**	**23.76**	**22.94**	**22.24**

Figure 2.22: The BSD68 and Set12 Datasets with Noise Levels $\sigma \in [5, 145]$ were used for testing, and the BSD400 Dataset was used for training. The Average PSNR (dB) of Different Blind Image Denoising Methods is presented below. The top and second-best results are **bolded and underlined**, respectively, in each column.

This work presents a wavelet-inspired invertible network (WINNet), which consists of sparsity-driven denoising networks and K levels of lifting-inspired invertible neural networks (LINNs). LINNs function as a non-linear redundant transform that can achieve flawless reconstruction, mimicking the features of wavelet transforms. The sparsity-driven denoising network adjusts to unseen noise levels by efficiently removing noise from the detail components of the transform coefficients in order to denoise images. Furthermore, blind WINNet demonstrates adaptability in picture deblurring tasks and performs exceptionally well in robust blind image denoising beyond the noise levels encountered during training, thanks to a model-inspired noise estimation network.

Chapter 3

Proposed Methodology

3.1 Method Overview

Inspiring from the paper [52] mentioned previously, we thought of finding out what other degradation except noise can possibly hinder the revelation process of the steganographic network. We evaluated the secret revealing performance of HiNet [18] against degradation like noise, blurring, sharpening, compression, etc. and found that it can't handle most of these. To make it robust against these corruptions, we took inspiration from yet another paper [39] that purposefully perturbs the stego image with corruptions to simulate real world degradation. We have designed our pipeline on the basis of these two ideas.

We trained our model initially for the steganography network. After 500 epochs, the model, which is an Invertible Neural Network gives satisfactory results in generating identical stego and can reveal accurate secrets. Next, we branched out the training procedure for handling different degradations. We simultaneously trained our initial model by adding different degradations to the stego image before training the model to reveal secrets from it. This way, the steganographic model works well even if the stego image becomes corrupted somehow during transmission.

We have shown three such degradation branches here in Figure 3.1. Every branch takes two images as input - a cover and a secret which pass through the forward concealing block that embeds the secret in the cover and generates the stego image, while some information shown in yellow blocks is lost in this process. The stego images then get passed in the backward concealing block, which recovers the secret image. The difference in the branches is during the transition between forward and backward passes, where we add degradation, specifically noise, blurring and sharpening with

the stego image. So the complete input of the entire model is two images- a cover and a secret—and the outputs are again two imaged- the recovered cover and the secret image.

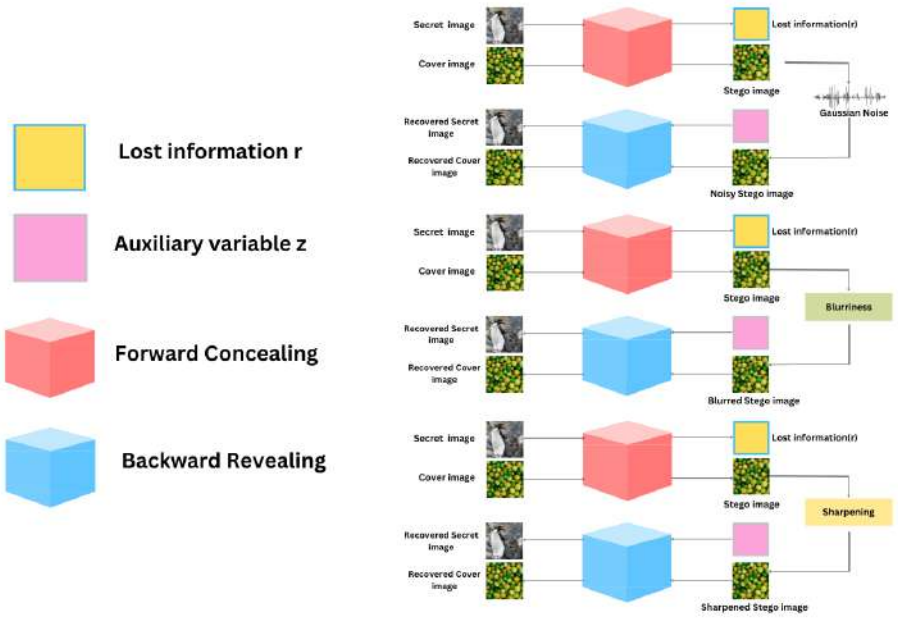


Figure 3.1: Our proposed pipeline

The workings of the blocks can be seen in detail below.

3.2 Concealing Block

In the forward concealing process, a secret image is embedded into a cover image using several concealing blocks to produce a stego image, along with any lost information.

3.3 Revealing Block

During the backward revealing process, the stego image and an auxiliary variable z drawn from a Gaussian distribution are input into a series of revealing blocks to retrieve the secret image. The concealing and revealing blocks consist of identical submodules and share the same network parameters, but the direction of information flow is reversed. directions.

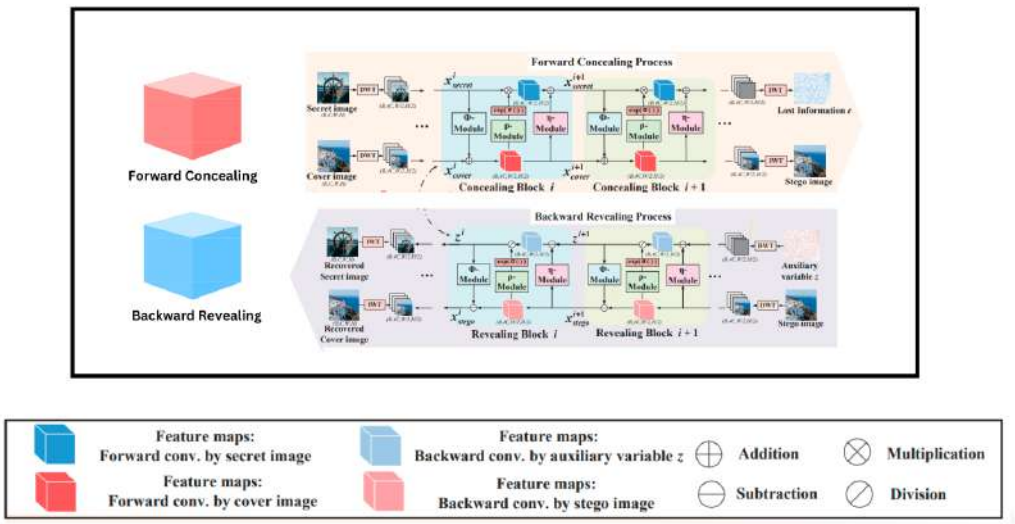


Figure 3.2: Concealing block and Revealing block of the INN

3.4 DWT

The Discrete Wavelet Transform (DWT) is crucial in both the forward concealing and backward revealing processes, enhancing the network's efficiency in managing image degradations. In the forward concealing process, the secret image x_{secret} and the cover image x_{cover} are decomposed into low and high-frequency wavelet sub-bands using DWT. This process divides the input images into multiple sub-bands, capturing various frequency components. Consequently, the feature map of size (B, C, H, W) is transformed into $(B, 4C, H/2, W/2)$, where B represents the batch size, C the number of channels, H the height, and W the width. This transformation captures detailed frequency information and reduces computational costs, speeding up the training process.

During the backward revealing process, the stego image x_{stego} and an auxiliary variable z undergo DWT, generating sub-bands that are processed through a series of revealing blocks. Afterward, these sub-bands are recombined using the Inverse Wavelet Transform (IWT) to reconstruct the secret image x_{secret}. This bidirectional use of wavelet transforms ensures accurate preservation and reconstruction of information.

Incorporating DWT offers several advantages. First, hiding images in the frequency domain, especially in the high-frequency sub-bands, is more effective than in the

pixel domain. This method reduces texture-copying artifacts and color distortion, enhancing hiding performance. By dividing the image into low and high-frequency sub-bands, the network can better embed the secret information into the cover image. Additionally, the perfect reconstruction property of wavelets ensures the original image information is well-preserved during transformation, minimizing information loss and improving the quality of both hidden and recovered images. The bidirectional symmetric nature of the wavelet transform allows seamless integration into HiNet's end-to-end training process without affecting its optimization and learning capabilities.

3.5 IWT

The Inverse Wavelet Transform (IWT) is primarily used to reconstruct images from their wavelet sub-bands. Here, IWT is employed during both the forward concealing and the backward revealing processes to ensure accurate and high-quality image restoration.

In the forward concealing process, after the secret and cover images have been decomposed into low and high-frequency wavelet sub-bands through Discrete Wavelet Transform (DWT), they are fed into a series of concealing blocks. The final outputs of these blocks are then processed by the IWT block to generate the stego image x_{stego} along with any lost information r. This step is essential as it ensures that the image, now containing hidden information, is reconstructed back to its original spatial dimensions with minimal loss in quality.

In the backward revealing process, the stego image x_{stego} and an auxiliary variable z undergo DWT to produce sub-bands that are input into revealing blocks. The revealing blocks work to recover the secret image x_{secret}. After passing through these blocks, the sub-bands are recombined using IWT to generate the recovered secret image x_{rec}. This process is vital for accurately extracting the hidden image without significant loss of detail or quality.

3.6 Enhancing HiNet Model to Handle Image Degradations

To enhance the robustness of the HiNet architecture, we implemented several strategies to address common image degradations, including noise addition, blurring, and

sharpening. Recognizing the limitations of the existing state-of-the-art (SotA) HiNet in handling these degradation scenarios, we integrated the following modifications into the Hinet framework:

1. **Training HiNet with Noisy Stego:** Following the forward pass of the HiNet model, upon obtaining the stego image, we introduced external Gaussian noise to induce corruption. Subsequently, the corrupted stego image was utilized for training. Through the backward pass, the model is now equipped to extract the secret from noisy stego images.

2. **Training HiNet with Blurred Stego:** Similarly, post-forward pass of the HiNet model, when presented with the stego image, we introduced external blurring to degrade the image quality. This blurred stego image was then employed for training purposes. Following the backward pass, the model demonstrates the capability to recover the secret from blurred stego images.

3. **Training HiNet with Sharpened Stego:** In this approach, subsequent to the forward pass of the HiNet model and acquisition of the stego image, external sharpening was applied to further distort the image. The sharpened stego image was used for subsequent training iterations. Through the backward pass, the model is now adept at extracting the secret from sharpened stego images.

Chapter 4

Results and Discussion

4.1 Dataset Used

The effectiveness and generalizability of our proposed methodology depend significantly on the quality and diversity of the dataset used during experimentation. As mentioned before, for our existing steganographic model, we use DIV2K (train) for training and DIV2K (test), COCO, ImageNet for testing.

In selecting an appropriate dataset for the degradation model, we prioritize characteristics that align with the specific requirements of our denoising modules considering the focus on steganographic image restoration. In the realm of image denoising, the SIDD (Smartphone Image Denoising Dataset) dataset has emerged as a widely accepted benchmark. This is made up of about 30,000 noisy photos taken with five typical smartphone cameras in ten distinct lighting scenarios to create their ground truth photographs. Its comprehensive nature makes it an ideal choice for evaluating the denoising capabilities of our proposed methodology.

4.2 Evaluation Metrics Used

In evaluating the performance of our enhanced HiNet model, we employed two widely recognized metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). These metrics were chosen due to their effectiveness in assessing image quality and their relevance to our research objectives.

In the context of image steganography, PSNR is crucial because it helps determine how much distortion the embedding process introduces into the cover image. Maintaining

a high PSNR ensures that the stego image remains visually similar to the original, which is important for the imperceptibility of the hidden data.

For steganographic applications, SSIM is particularly valuable because it aligns more closely with human visual perception. Ensuring a high SSIM value means that the stego image is not only visually similar but also structurally consistent with the original image, which is critical for detecting distortions that might reveal the presence of hidden data.

Both PSNR and SSIM provide complementary insights into the quality of the stego images. While PSNR offers a clear numerical value indicating overall error, SSIM provides a perceptual quality measure that accounts for human visual perception. Together, they ensure a comprehensive evaluation of image quality. Moreover, using PSNR and SSIM allows us to benchmark our enhanced HiNet model against existing state-of-the-art methods in a standardized manner. These metrics are commonly used in the literature, facilitating a direct comparison of performance improvements.

4.3 Conducted Experiments

4.3.1 Experimental Setup

4.3.2 Tuned Hyperparameters

Batch Size Adjustment

The primary objective of adjusting the batch size was to address the memory overflow issues observed during training sessions, which can hinder the training process and lead to suboptimal model performance.

- Original Batch Size: 16
- Adjusted Batch Size: 8

Reducing the batch size effectively mitigated memory overflow problems, ensuring smoother and more efficient training sessions. By halving the batch size, we were able to fit the model within the available memory constraints, which allowed for more stable training iterations without encountering memory-related errors. This adjustment was crucial in maintaining the integrity of the training process, preventing interruptions and ensuring consistent progress towards model convergence.

Learning Rate Adjustment

The objective of adjusting the learning rate was to determine an optimal rate that would minimize the loss function effectively during training while preventing model divergence. An inappropriate learning rate can either slow down the training process or lead to unstable training, causing the model to diverge.

- Original Learning Rate: $-10^{-4.5}$
- Adjusted Learning Rate: -10^{-5}

Through iterative adjustments of the learning rate and close monitoring of the training performance, we identified an optimal learning rate that facilitated stable convergence of the loss function. This fine-tuning process aimed to balance training stability and convergence speed. A lower learning rate was found to promote more stable training dynamics, reduce the risk of overshooting minima, and ultimately lead to improved model robustness. This adjustment was essential in achieving a balance between rapid convergence and maintaining the stability of the training process.

4.3.3 Degradations Considered

We tested HiNet and found that it can't handle degradation like noise, blurring, sharpening and compression. So we looked into how we can solve this limitation of the existing model. From our idea of StegaStamp, we thought adding perturbations to the model can make it perform better with real world degradations.

We trained HiNet with simple modifications like adding different degradations to the STego image generated in the forward pass before passing it to the backward pass. We specifically worked with three degradations - noise, blurring, sharpening. In all cases our modified model is seen to perform better than the original Hinet model.

Image Degradation Parameters

- **Blurring** Blurring was applied to the stego images to simulate typical image degradation scenarios. The Gaussian kernel, with a specified size of 5×5 and a sigma value of 10, was employed to introduce blurring effects. This approach was integral in training the model to recover hidden information from images that have undergone blurring, thus enhancing the robustness of the HiNet architecture against such degradation.
- **Sharpening** Sharpening was incorporated to evaluate the model's proficiency in handling image enhancements that amplify contrast and edge definition. The sharpening process utilized a Laplacian kernel, characterized by the matrix:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

with a kernel size of 3×3 and an alpha value of 1.5 to control the intensity of sharpening. By applying this kernel, sharpened images were produced for training purposes. This procedure enabled the model to learn how to effectively extract concealed information from images subjected to sharpening, thereby improving its adaptability and performance in scenarios involving enhanced image features.

- **Noise** We used random Gaussian noise for training. The `randn()` function from the NumPy library generates random samples from a normal (Gaussian) distribution, which were then applied to the images. Introducing noise was crucial for simulating real-world scenarios where images are often subject to various forms of noise during transmission or storage.

4.4 Analysis

4.4.1 Quantitative Analysis

Table 4.1: Average PSNR Comparison with HiNet

	Noise	Blur	Sharpening	JPEG Compression
HiNet	7.11	14.773	13.519	13.07
Ours	19.71	32.981	41.552	-

Table 4.2: Average SSIM Comparison with HiNet

	Noise	Blur	Sharpening	JPEG Compression
HiNet	0.009	0.534	0.527	0.41
Ours	0.37	0.91	0.9873	-

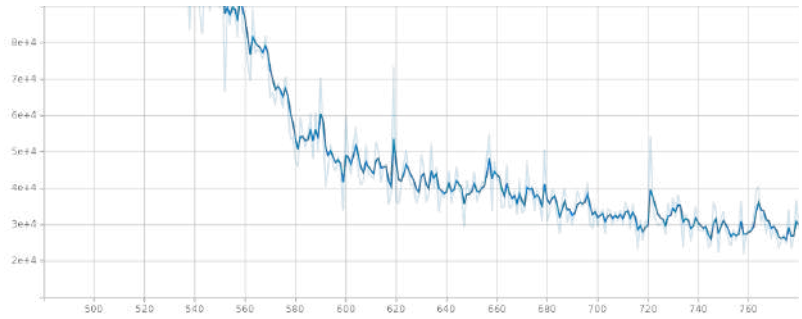


Figure 4.1: Ours - Training Loss vs Epoch for Image Noising

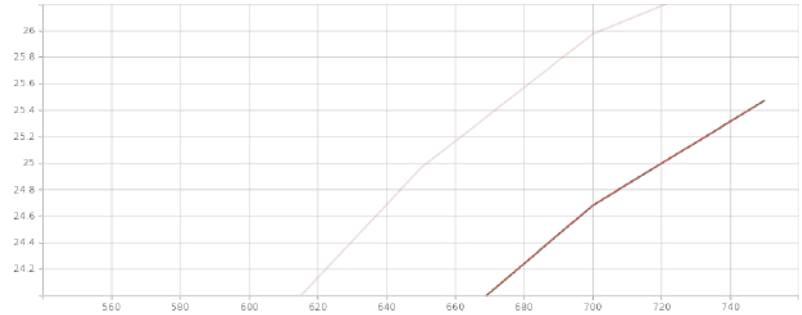


Figure 4.2: Ours - $PSNR_S$ vs Epoch for Image Noising

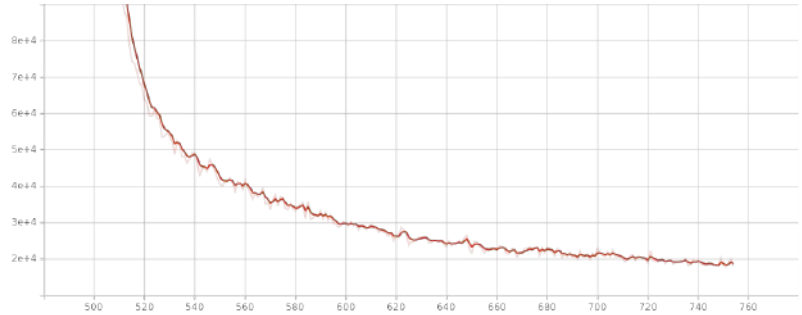


Figure 4.3: Ours - Training Loss vs Epoch for Image Sharpening

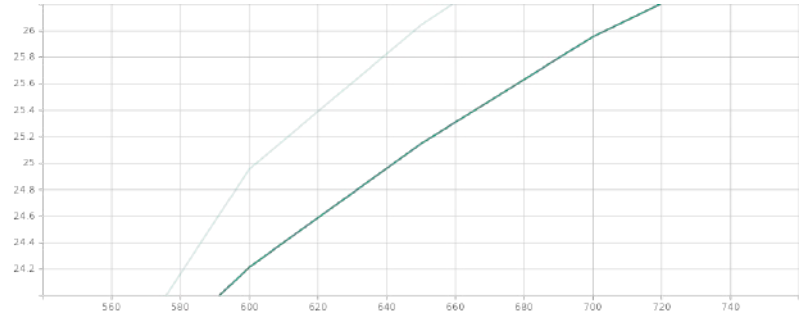


Figure 4.4: Ours - $PSNR_S$ vs Epoch for Image Sharpening

4.4.2 Qualitative Analysis

When stego image is corrupted with noise

We trained the SoTA HiNet with external Gaussian noise and then after testing we got better results than the original HiNet. When the original HiNet was tested with

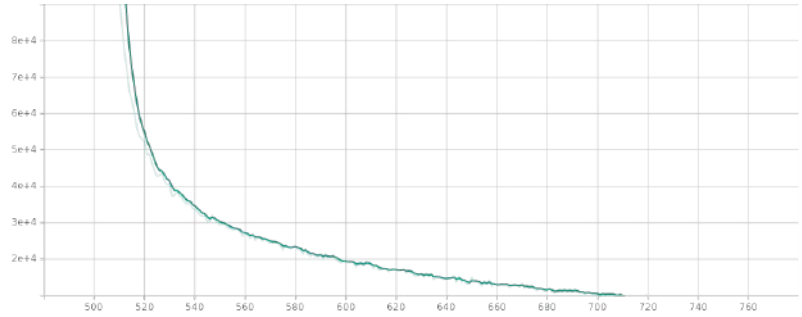


Figure 4.5: Ours - Training Loss vs Epoch for Image Blurring

noises there was no traces of secret in it. Now after our training we can see the secret image to an extent.

When stego image is corrupted with blurring

We trained the SoTA HiNet with blurriness and then after testing we got better results than the original HiNet. When the original HiNet was tested with blurriness there were no color channels and we got only traces of the cover image. Now after our training we can see the secret image to an extent.

When stego image is corrupted with sharpening

We trained the SoTA HiNet with sharpening and then after testing we got better results than the original HiNet. When the original HiNet was tested with sharpening, we got the secret but there was huge contrast and distortions in the retrieved secret. Now after our training we can see there is less contrast and the retrieved secret is much better now.

When stego image is corrupted with compression

When original HiNet was tested with image compression techniques the secret image was completely lost and the HiNet failed miserably.

Table 4.3: Gaussian noise $\sigma = 10$

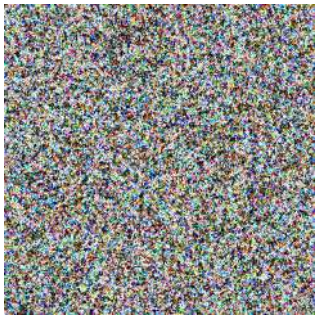
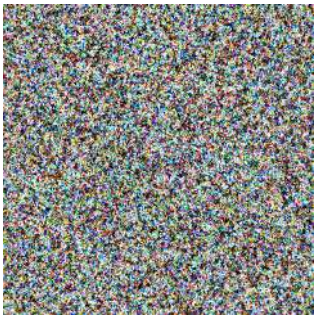
Cover		
Secret		
Stego		
Revealed Secret - HiNet		
Revealed Secret - Ours		

Table 4.4: Blurring with $\sigma = 10$ and kernel size $= 5$

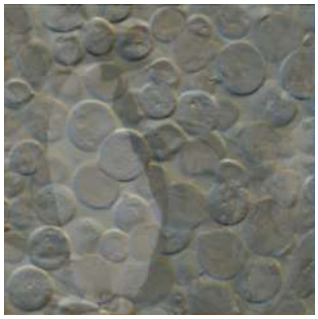
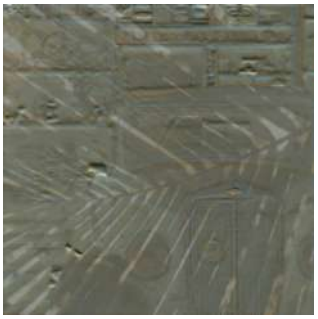
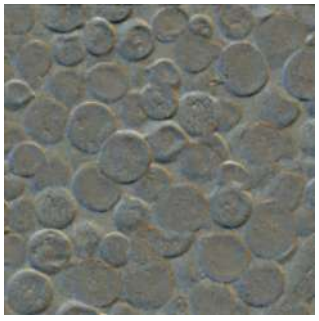
Cover		
Secret		
Stego		
Revealed Secret - HiNet		
Revealed Secret - Ours		

Table 4.5: Sharpening with $\alpha = 1.5$ and kernel size $= 3$

Cover		
Secret		
Stego		
Revealed Secret - HiNet		
Revealed Secret - Ours		

Table 4.6: JPEG Compression QF = 90

Cover	
Secret	
Stego	
Revealed Secret - HiNet	

Chapter 5

Conclusion

5.1 Summary

In our pursuit to enhance the robustness of the state-of-the-art architecture HiNet against various image degradations, our endeavors have yielded promising outcomes. Through rigorous experimentation and training iterations, we succeeded in achieving superior performance across different degradation scenarios, notably in noise reduction, sharpening, and blurring tasks.

Our achievements signify a notable advancement in the adaptability and resilience of HiNet, positioning it as a versatile solution capable of delivering high-quality results even in challenging conditions. By leveraging innovative techniques and fine-tuning model parameters, we were able to elevate the efficacy of HiNet in mitigating the adverse effects of image degradations, thereby improving its practical utility across diverse applications.

5.2 Future scope

Looking ahead, our focus extends to further extending the robustness of HiNet by addressing additional image degradation factors, such as JPEG compression and other forms of distortion. Through continued experimentation and refinement, we aim to fortify HiNet's performance across a broader spectrum of real-world scenarios, ensuring its relevance and effectiveness in addressing contemporary challenges in image processing and computer vision. Moreover for a generic image with unknown degradation as input, we plan to make an ensemble of all the revealed secrets that we get from the different trained networks. We plan to deal with the noise present in the fi-

nal ensembled image by adding a denoising module after the ensemble, i.e., the input of the denoiser will be the output of the ensemble.

In essence, our accomplishments underscore the potential of HiNet as a foundational framework for advancing the field of image restoration, with implications for diverse domains ranging from medical imaging to surveillance and beyond. As we embark on future endeavors, we remain committed to pushing the boundaries of innovation and driving meaningful progress in the quest for robust and reliable image processing solutions.

References

[1] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[2] S. Baluja, "Hiding images in plain sight: Deep steganography," *Advances in neural information processing systems*, vol. 30, 2017.

[3] S. Baluja, "Hiding images within images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1685–1697, 2019.

[4] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE transactions on image processing*, vol. 10, no. 5, pp. 783–791, 2001.

[5] T. Bui, S. Agarwal, N. Yu, and J. Collomosse, "Rosteals: Robust steganography using autoencoder latent space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 933–942.

[6] H. Chen, L. Song, Z. Qian, X. Zhang, and K. Ma, "Hiding images in deep probabilistic models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 776–36 788, 2022.

[7] X. Deng, C. Gao, and M. Xu, "Pirnet: Privacy-preserving image restoration network via wavelet lifting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 368–22 377.

[8] J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.

[9] S. Ghamizi, M. Cordy, M. Papadakis, and Y. Le Traon, "Evasion attack steganography: Turning vulnerability of machine learning to adversarial attacks into a real-world application," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 31–40.

[10] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," *Advances in neural information processing systems*, vol. 30, 2017.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[12] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *2012 IEEE International workshop on information forensics and security (WIFS)*, IEEE, 2012, pp. 234–239.

[13] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, pp. 1–13, 2014.

[14] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on image processing*, vol. 8, no. 1, pp. 58–68, 1999.

[15] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE access*, vol. 6, pp. 38 303–38 314, 2018.

[16] J. Jia, Z. Gao, K. Chen, *et al.*, "Rihoop: Robust invisible hyperlinks in offline and online photographs," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7094–7106, 2020.

[17] J. Jia, Z. Gao, D. Zhu, X. Min, G. Zhai, and X. Yang, "Learning invisible markers for hidden codes in offline-to-online photography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2273–2282.

[18] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, "Hinet: Deep image hiding by invertible network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 4733–4742.

[19] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[21] D. Kim, C. Shin, J. Choi, D. Jung, and S. Yoon, "Diffusion-stego: Training-free diffusion generative steganography via message projection," *arXiv preprint arXiv:2305.18726*, 2023.

[22] V. Kishore, X. Chen, Y. Wang, B. Li, and K. Q. Weinberger, "Fixed neural network steganography: Train the images, not the network," in *International Conference on Learning Representations*, 2021.

[23] D. Lerch-Hostalot and D. Megías, "Unsupervised steganalysis based on artificial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.

[24] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International conference on image processing (ICIP)*, IEEE, 2014, pp. 4206–4210.

[25] J. Li, K. Niu, L. Liao, *et al.*, "A generative steganography method based on wgan-gp," in *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part I 6*, Springer, 2020, pp. 386–397.

[26] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 897–911, 2020.

[27] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[28] M.-m. Liu, M.-q. Zhang, J. Liu, Y.-n. Zhang, and Y. Ke, "Coverless information hiding based on generative adversarial networks," *arXiv preprint arXiv:1712.06951*, 2017.

[29] X. Liu, Z. Ma, J. Ma, J. Zhang, G. Schaefer, and H. Fang, "Image disentanglement autoencoder for steganography without embedding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 2303–2312.

[30] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, "Large-capacity image steganography based on invertible neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 816–10 825.

[31] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on lsb matching revisited," *IEEE Transactions on information forensics and security*, vol. 5, no. 2, pp. 201–214, 2010.

[32] Q. Nguyen, T. Vu, C. Pham, A. Tran, and K. Nguyen, "Stable messenger: Steganography for message-concealed image generation," *arXiv preprint arXiv:2312.01284*, 2023.

[33] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12*, Springer, 2010, pp. 161–177.

[34] R. Rahim, S. Nadeem, *et al.*, "End-to-end trained cnn encoder-decoder networks for image steganography," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.

[35] J. Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proceedings of 3rd IEEE International Conference on Image Processing*, IEEE, vol. 3, 1996, pp. 239–242.

[36] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[37] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2015.

[38] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "Ssgan: Secure steganography based on generative adversarial networks," in *Advances in Multimedia Information Processing–PCM 2017: 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28-29, 2017, Revised Selected Papers, Part I 18*, Springer, 2018, pp. 534–544.

[39] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2117–2126.

[40] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[41] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," in *Twelfth international conference on machine vision (ICMV 2019)*, SPIE, vol. 11433, 2020, pp. 991–1005.

[42] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "Sstegan: Self-learning steganography based on generative adversarial networks," in *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part II 25*, Springer, 2018, pp. 253–264.

[43] P. Wei, S. Li, X. Zhang, G. Luo, Z. Qian, and Q. Zhou, "Generative steganography network," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1621–1629.

[44] P. Wei, G. Luo, Q. Song, X. Zhang, Z. Qian, and S. Li, "Generative steganographic flow," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2022, pp. 1–6.

[45] P. Wei, Q. Zhou, Z. Wang, Z. Qian, X. Zhang, and S. Li, "Generative steganography diffusion," *arXiv preprint arXiv:2305.03472*, 2023.

[46] X. Weng, Y. Li, L. Chi, and Y. Mu, "High-capacity convolutional video steganography with temporal residual modeling," in *Proceedings of the 2019 on international conference on multimedia retrieval*, 2019, pp. 87–95.

[47] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1515–1524.

[48] P. Wu, Y. Yang, and X. Li, "Stegnet: Mega image steganography capacity with deep convolutional network," *Future Internet*, vol. 10, no. 6, p. 54, 2018.

[49] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7875–7884.

[50] C. Yu, D. Hu, S. Zheng, W. Jiang, M. Li, and Z.-q. Zhao, "An improved steganography without embedding based on attention gan," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1446–1457, 2021.

[51] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[52] J. Yu, X. Zhang, Y. Xu, and J. Zhang, "Cross: Diffusion model makes controllable, robust and secure image steganography," *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[53] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 223–10 234, 2020.

[54] K. A. Zhang, A. Cuesta-Infante, and K. Veeramachaneni, "Steganogan: High capacity image steganography with gans," *arXiv preprint arXiv:1901.03892*, 2019. [Online]. Available: `https://arxiv.org/abs/1901.03892`.

[55] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia tools and applications*, vol. 78, pp. 8559–8575, 2019.

[56] Z. Zhang, G. Fu, R. Ni, J. Liu, and X. Yang, "A generative method for steganography by cover synthesis with auxiliary semantics," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 516–527, 2020.

[57] Z. Zhou, Y. Su, J. Li, *et al.*, "Secret-to-image reversible transformation for generative steganography," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[58] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," in *Cloud Computing and Security: First International Conference, ICCCS 2015, Nanjing, China, August 13-15, 2015. Revised Selected Papers 1*, Springer, 2015, pp. 123–132.

[59] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.