

Low Resolution Coverless Image Steganography for High Resolution Images using Generative Adversarial Networks

H G D S Aiyasinghe

2025



Low Resolution Coverless Image Steganography for High Resolution Images using Generative Adversarial Networks

Author

H G D S Ariyasinghe

Index Number: 20000146

Supervisor: Dr. Kasun Gunawardana

Co-Supervisor: Dr. Tharindu Wijethilake

June 27, 2025

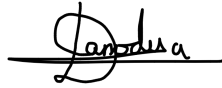
Submitted in partial fulfillment of the requirements of the B.Sc in Computer
Science Final Year Project (SCS4224)



Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: H G D S Ariyasinghe



Signature of Candidate

Date: 2025/06/30

This is to certify that this dissertation is based on the work of Mr. H G D S Ariyasinghe under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor's Name: Dr. Kasun Gunawardana



Signature

Date: 30-June-2025

Co-Supervisor's Name: Mr. Tharindu Wijethilake



2025-06-30

Signature

Date:

Abstract

Traditional image steganography techniques modify cover images to hide information, making them vulnerable to detection. Coverless image steganography eliminates this dependency but often struggles to maintain high reconstruction fidelity while ensuring robust concealment. This study proposes a novel coverless image steganography method using Generative Adversarial Networks (GANs) to address these limitations. GANs excel at learning complex data distributions and generating realistic stego images that effectively encode high-resolution secret images, even after aggressive downscaling. By downscaling secret images prior to embedding, our approach achieves a fourfold increase in embedding capacity while reducing computational burden and maintaining competitive reconstruction quality as measured by SSIM, PSNR, and MAE metrics. Extensive experiments demonstrate superior performance in embedding capacity, reconstruction fidelity, and concealment robustness compared to traditional multi-image steganography techniques. This research provides an effective solution for secure high-resolution image embedding, advancing coverless image steganography.

Contents

Abstract	2
Chapter 01	1
1.1 Introduction	1
1.2 Motivation	3
1.3 Problem Statement	3
1.4 Research Questions	4
1.5 Aims and Objectives	5
1.6 Significance of the Project	5
1.7 Scope	6
1.7.1 In Scope	6
1.7.2 Out of Scope	6
1.8 Research Methodology and Evaluation Plan	7
1.8.1 Research Methodology	7
1.8.2 Evaluation Plan	8
1.9 Project Timeline	8
Chapter 02	10
2.1 Literature Review	10
2.1.1 Theories	10
2.1.1.1 Convolutional Neural Networks	10
2.1.1.2 Generative Adversarial Networks	12
2.1.2 Steganography	13
2.1.2.1 Overview	13
2.1.2.2 Cover Modification-based Steganography	14
2.1.2.3 Cover Selection-based Steganography	15
2.1.2.4 Cover Synthesis-based Steganography	15
2.1.3 Steganalysis	16
2.1.4 Research Gap	17
Chapter 03	19

3.1	Research Design	19
3.1.1	Step 1: Defining of Research Questions	21
3.1.2	Step 2: Literature Review	21
3.1.3	Step 3: Data Collection	21
3.1.4	Step 4: High-Resolution Restoration Model	22
3.1.5	Step 5: High-Resolution Restoration Model Evaluation	23
3.1.6	Step 6: Stego Image Generation Model	23
3.1.7	Step 7: Stego Image Generation Model Evaluation	24
3.1.8	Step 8: Model Integration	24
3.1.9	Step 9: Integrated Model Evaluation	24
Chapter 04		25
4.1	Model Architectures and Experimental Results	25
4.1.1	High-Resolution Restoration Model Experiments	25
4.1.1.1	Experiment 1:High-Resolution Image Restoration Using Interpolation Techniques.	26
4.1.1.2	Experiment 2:Auto-Encoder Based High-Resolution Image Restoration.	31
4.1.1.3	Experiment 3: Bicubic Interpolation Enhanced with Refinement Network for High-Resolution Image Reconstruction.	35
4.1.2	Stego Image Generation Model	39
4.1.2.1	Generator and Discriminator Architectures	40
4.1.2.2	Customized Loss Function for Stego Image Generation and Enhancement	44
4.1.2.3	Experiment 1: Stego Image Generation with Same Resolution as Secret Image	47
4.1.2.4	Experiment 2: Low-Resolution Stego Image Generation Using Bicubic Interpolation	52
4.1.2.5	Experiment 3: Auto-Encoder-Based Low-Resolution Stego Image Generation	57
4.1.2.6	Experiment 4: Scrambling-Based Concealment Prior to Stego Generation	61
4.1.2.7	Selection of Final Models for Evaluation	68
Chapter 05		70
5.1	Comprehensive Evaluation of Final Stego Image Generation Models	70

5.1.1	Qualitative Evaluation: Visual Analysis of Reconstructed Im-	
	ages	71
5.1.1.1	Quantitative Evaluation: SSIM, PSNR, and MAE	71
5.1.1.2	RGB Histogram Comparison Between Original and	
	Reconstructed Secret Images	72
5.1.1.3	Lab Histogram Comparison Between Original and	
	Reconstructed High-Resolution Secret Images	74
5.1.1.4	Histogram Visualization for RGB and Lab Channels	76
5.1.2	Evaluation of Hiding Capacity and Secret Image Size Reduction	78
5.1.3	Evaluation Against Steganalysis Tools	79
5.1.4	Model Applicability	80
5.2	Implementation Details and Experimental Setup	82
5.2.1	Datasets	82
5.2.2	Programming Tools and Libraries	82
5.2.3	Training Configuration and Experimental Setup	83
Chapter 06		84
6.1	Conclusion	84
6.1.1	Conclusion About Research Questions	84
6.1.2	Conclusion About Research Problem	85
6.1.3	Limitations	86
6.1.4	Implications for Further Research	86

List of Figures

1.1	The basic process of traditional image steganography	2
1.2	The basic process of coverless image steganography	2
1.3	Methodology in high level	7
1.4	Gantt chart of the research project	9
2.1	General architecture of a Convolutional Neural Network (Convolutional Neural Network (CNN)) (Ayub[2022]).	12
2.2	Architecture of a Generative Adversarial Network	13
3.1	Research Design	20
3.2	Overview of High-Resolution Restoration Model	22
3.3	Overview of Stego Image Generation Model	23
3.4	Overview of Integrated Model	24
4.1	Bilinear Interpolation Results	27
4.2	Bicubic Interpolation Results	28
4.3	Nearest Neighbor Interpolation Results	30
4.4	Auto-Encoder Architecture for High-Resolution Image Restoration	32
4.5	Hyperparameter Tuning Process for the Loss Function	33
4.6	Auto-Encoder Results	34
4.7	Bicubic Interpolation Enhanced with Refinement Network	35
4.8	Refinement Network Architecture for High-Resolution Image Restoration	36
4.9	Hyperparameter Tuning Process for the Loss Function	37
4.10	Bicubic Interpolation Enhanced with Refinement Network Results	38
4.11	CycleGAN Generator Architecture	41
4.12	PatchGAN Discriminator Architecture	42
4.13	U-Net Generator Architecture	43
4.14	High-Level Architecture of Stego Image Generation and Reconstruction Pipeline for Experiment 1.	48
4.15	Qualitative results for Experiment 1 showing the Secret Image, Stego Image, Reconstructed Secret Image, and Refined Reconstructed Secret Image for several test cases.	50

4.16 High-Level Architecture for Experiment 2: Low-Resolution Stego Image Generation Pipeline Incorporating Bicubic Interpolation.	53
4.17 Visual results from Experiment 2. Each row corresponds to one test sample, progressing from the original HR secret image to the final refined reconstruction.	55
4.18 Examples demonstrating limitations of Experiment 2 for simple secret images. The stego images in these cases still reveal perceptual hints of the secret content.	56
4.19 Comparison between high-resolution secret image and the low-resolution image generated via auto-encoder. Note the significant loss of visual detail, which enhances concealment.	57
4.20 High-Level Architecture of Auto-Encoder-Based Low-Resolution Stego Image Generation Pipeline (Experiment 3).	59
4.21 Experiment 3 Results. Columns from left to right: High-Resolution Secret Image, Auto-Encoded LR Image, Stego Image, Reconstructed LR Image, Reconstructed HR Image, Refined HR Image.	60
4.22 Visual output of a sample scrambled image.	62
4.23 High-Level Architecture of the Scrambling-Based Stego Generation and Reconstruction Pipeline (Experiment 4).	65
4.24 Experiment 4 Results. Columns from left to right: HR Secret Image, LR Secret Image, Scrambled LR Image, Stego Image, Reconstructed Scrambled LR Image, Inverse-Scrambled LR Image, Reconstructed HR Image, Refined HR Image.	67
5.1 Sample Image Used for Histogram Comparison.	76
5.2 RGB Histogram Comparison Between Original and Reconstructed Secret Images (Experiment 2 vs Experiment 4).	77
5.3 Lab Histogram Comparison Between Original and Reconstructed Secret Images (Experiment 2 vs Experiment 4).	77

List of Tables

4.1	SSIM, PSNR, and MSE values for Images	27
4.2	SSIM, PSNR, and MSE values for Images	29
4.3	SSIM, PSNR, and MSE values for Images	30
4.4	SSIM, PSNR, and MSE values for Images across Bilinear, Bicubic, and Nearest Neighbour Interpolation	31
4.5	SSIM, PSNR, and MSE values for Images	34
4.6	SSIM, PSNR, and MSE values for Images	38
4.7	Combined SSIM, PSNR, and MSE values for Images across different experiments: Bicubic Interpolation, Auto-Encoder, and Bicubic with Refinement Network	39
5.8	Quantitative Comparison of Reconstruction Quality	72
5.9	Quantitative Comparison of Reconstruction Quality with Prior Work	72
5.10	RGB Histogram Comparison Between Original and Reconstructed Secret Images	74
5.11	Lab Histogram Comparison Between Original and Reconstructed High- Resolution Secret Images	75
5.12	Comparison of Embedding Capacity with Hiding Capacity Factor	79
5.13	Training Hyperparameters	83

List of Acronyms

CIS	Coverless Image Steganography
CNN	Convolutional Neural Network
DCGAN	Deep Convolutional Generative Adversarial Networks
DCT	Discrete Cosine Transformation
DFT	Discrete Fourier Transformation
DWT	Discrete Wavelet Transformation
GAN	Generative Adversarial Network
GANs	Generative Adversarial Networks
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LSB	Least Significant Bit
MSE	Mean Squared Error
POVs	Pairs of Values
PSNR	Peak Signal-to-Noise Ratio
ReLU	Rectified Linear Unit
SGAN	Steganographic GAN
SGD	Stochastic Gradient Descent
SSGAN	Secure Steganographic GAN
SSIM	Structural Similarity Index
SURF	Speeded-Up Robust Features
WGAN	Wasserstein Generative Adversarial Network

Chapter 01

1.1 Introduction

Due to advancements in technology in recent years, secure data transmission across network channels has become crucial. As a solution for secure data transmission, cryptography was introduced. In cryptography the data is encrypted using an encryption key such that the original data remains invisible, while ciphertext is perceptible to the human eye. Upon reaching the destination, the ciphertext is converted back in to original data using a decryption key (van Leeuwen 1990). In this case people can discover that secret data is going through the channel. In this context, a novel research area called steganography, has introduced. It encompasses techniques capable of introducing secret information within media in a manner that remains undetectable to third parties. Unlike cryptography, which focuses on rendering information unreadable through various operations, steganography aims to achieve invisibility, making the secret data imperceptible to unintended recipients.

The term steganography is derived from two Greek words, 'stegos' and 'grafia', meaning 'cover' and 'writing' respectively. Modern steganography uses digital media such as images, audio and video to hide secret data.

Image steganography is one of the major area in steganography in which secret data is hidden inside an image. Here secret data can be a text, image, or multiple images etc. In traditional image steganography, secret data embedded within the pixels of an image which is known as cover image. The embedded image is called the stego-image or carrier image. The main intention is to hide the secret information in such a way that the stego-image and the cover image are visually indistinguishable to human eye. The basic process of traditional image steganography is shown in Figure 1.1. With the advancements in steganography, steganalysis tools have also become more powerful. Hence, traditional image steganography methods have become vulnerable to modern steganalysis tools. As a solution, researchers found a novel coverless image steganography method, which is capable of generating stego-images directly from secret data without embeddings. Detection of the presence of

a secret message becomes challenging in this approach, making coverless steganography more resistant to steganalysis tools. Absence of cover image makes it more secure compared to cover image steganography. The basic process of coverless image steganography is shown in Figure 1.2.

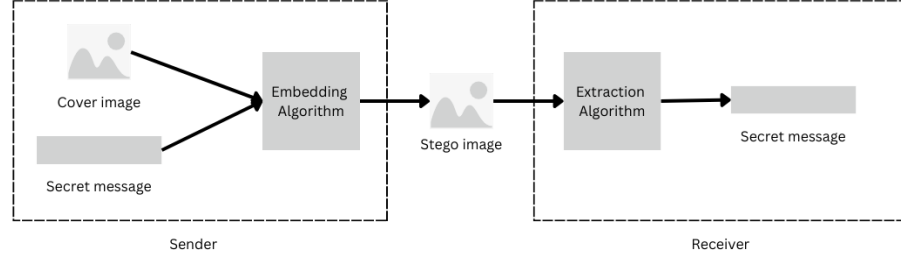


Figure 1.1: The basic process of traditional image steganography

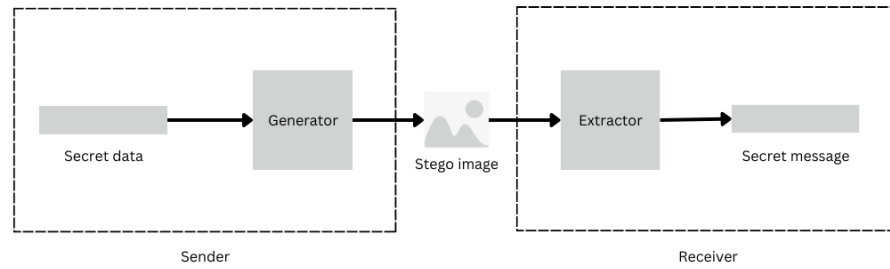


Figure 1.2: The basic process of coverless image steganography

Steganography methods can be divided into two based on their domain named, spatial and transformation domain. Spatial domain based methodologies directly modify image data by working on individual pixel values. It changes the pixel's bit sequence in order to hide secret data such that changes are not visible to human eye. One of such algorithm is the Least Significant Bit (**LSB**) proposed by (Neeta et al. 2006). **LSB** algorithm is the most widely used spatial domain steganography algorithm used earlier. It works by modifying the least significant bit of the image's pixel value. Transformation domain techniques, operate by transforming the cover image to the frequency domain and then embeds the secret data into it by changing the frequencies. After embedding the secret data, the transformed image is then converted again to spatial domain to visualize the image. In literature, there are many transformation domain methods were implemented using different kinds transformations such as Discrete Cosine Transformation (**DCT**) (Watson et al. 1994), Discrete Wavelet Transformation (**DWT**) (Stanković & Falkowski 2003) and Discrete Fourier Transformation (**DFT**) (Paulson 2006).

Steganography and steganalysis are two critical, intertwined research domains. While steganography seeks to hide information, steganalysis aims to uncover it. Employing diverse methods like visual and statistical analysis, steganalysis not only detects hidden messages but also plays a pivotal role in developing more secure steganographic models by employing a steganalysis model as a component and also, identifying the most suitable steganalysis approach to evaluate their effectiveness.

1.2 Motivation

As highlighted in [Liu, Ke, Zhang, Lei, Li, Zhang & Yang \(2020\)](#), traditional image steganographic methods face two common limitations of achieving high hiding capacity and ensuring security against sophisticated steganalysis tools like Xu-Net proposed by [Xu et al. \(2016\)](#) and SRNet proposed by [Boroumand et al. \(2019a\)](#). When exploring the literature, we can observe how the utilization of GANs in steganographic models has proved to improve upon these limitations by enhancing the hiding capacity as well as its security as the stego-image is generated artificially.

Thereby, the motivation behind this research is to extend Generative Adversarial Network ([GAN](#)) based models to address the problem of low resolution coverless image steganography for high resolution images while improving the shortcomings of recovery accuracy limitations found within existing coverless image steganographic models.

1.3 Problem Statement

As the need for information hiding increases, researchers have undertaken many new approaches in steganography. In the context of Coverless Image Steganography (CIS), a stego image is typically generated directly from the secret image. Current image steganography techniques focus on generating stego images that maintain the same resolution as the original secret images. This process involves a communication protocol between the sender and receiver, where the sender generates the stego image using the secret image and sends it through a communication channel. Upon receiving the stego image, the receiver then reconstructs the secret image.

While existing CIS methods are effective for embedding and concealing data, they are often constrained by the high storage capacity and bandwidth requirements associated with high-resolution images. These limitations hinder the practical application of steganography in scenarios where storage and bandwidth are restricted

or costly.

Furthermore, all aforementioned steganography methods do not adequately address the need for generating low-resolution stego images from high-resolution secret images. This capability is crucial for optimizing storage and network bandwidth utilization, as well as enhancing resistance to steganalysis tools that detect hidden data.

With the advancements in [GAN](#), researchers tend to incorporate it for image steganography. The reason is [GAN](#) can generate high-quality images that closely resemble natural images. This makes it difficult for steganalysis tools to distinguish between steganographic images and regular images, thereby enhancing the imperceptibility.

To address this gap, this research proposes to explore coverless image steganography for high-resolution images by employing Generative Adversarial Networks ([GAN](#)). The objective is to develop a novel steganography method that generates low-resolution stego images, thereby improving efficiency in storage and transmission, and increasing resistance to steganalysis while maintaining the integrity and security of the hidden data.

In addition, this high-to-low resolution coverless image steganography approach aims to achieve high hidden capacity, a feature that was previously attempted using multi-image steganography techniques. However, those methods often suffered from poor reconstruction quality of the secret image and relied heavily on cover embedding techniques, compromising the essence of coverless steganography ([Das et al. 2021a](#)). By contrast, the proposed method leverages GANs to overcome these limitations, offering a more robust and efficient solution for secure data hiding.

1.4 Research Questions

1. How can Generative Adversarial Networks ([GAN](#)) be utilized to generate low-resolution stego images from high-resolution secret images for coverless image steganography?
2. What are the most effective Generative Adversarial Network ([GAN](#)) architectures and techniques for embedding data in low-resolution stego images to maximize data capacity and minimize perceptual distortion?
3. How does the reduction in resolution of stego images impact the detectability of hidden data by standard steganalysis tools?

1.5 Aims and Objectives

The aim of this research is to design and develop a novel coverless steganography method capable of successfully generating low-resolution stego images from high-resolution secret images, while enabling better reconstruction of the original secret image.

The objectives of this research are as follows:

1. Identify and analyse existing image steganography methods and their architectures.
2. Investigate and select an appropriate **GAN** architecture and techniques for generating low-resolution stego images.
3. Design and develop a novel coverless image steganography method that generates low-resolution stego images from high-resolution secret images.
4. Determine if the implemented model could resist modern steganalysis tools.

1.6 Significance of the Project

In today's world, where data privacy and security are major concerns, developing advanced covert communication techniques is crucial. Image steganography remains a prominent and evolving research field within computer science. Currently, many steganographic models have achieved success in hiding information effectively. However, despite recent advancements, Coverless Image Steganography (**CIS**) has yet to fully realize its potential, especially when dealing with high-resolution images. This presents a valuable opportunity to explore and develop new approaches capable of efficiently handling high-resolution data.

By addressing the challenges associated with high storage demands and bandwidth limitations, such methods can enable the practical use of **CIS** for secure image-based communication. Furthermore, this research aims not only to optimize storage and bandwidth usage but also to increase the hidden capacity within stego images. Enhancing the embedding capacity is vital for transmitting more information securely without compromising the quality or detectability of the stego image. Achieving this would significantly improve the applicability of **CIS** in real-world scenarios, especially in environments where resource efficiency and data confidentiality are paramount.

This research will investigate the limitations of current **CIS** methods and explore various approaches to handling high-resolution images. It will also examine how to apply **GAN** to design a model capable of generating low-resolution stego images from high-resolution inputs. Developing such a model will significantly contribute to the scientific community and can be applied across various fields for secure data transmission. Furthermore, this model could serve as a stepping stone for developing more sophisticated **CIS** models.

1.7 Scope

1.7.1 In Scope

1. Develop and implement a novel **CIS** method using **GAN** to generate low-resolution stego images from high-resolution secret images.
2. Investigate and implement methods to obtain low-resolution images from high-resolution images. This includes downscaling techniques and algorithms that preserve essential details needed for later reconstruction.
3. Conduct experiments to validate the effectiveness of the developed method specifically for high resolution images.
4. Train the implemented model and evaluate its results. This includes testing the method's performance in terms of data capacity, perceptual quality of the stego images, resistance to steganalysis tools, success rate of data extraction, and accuracy of high-resolution image reconstruction.

1.7.2 Out of Scope

1. This project focuses exclusively on coverless image steganography, excluding cover modification and cover selection methods.
2. This project will not investigate the applicability of other deep learning models during implementation.

1.8 Research Methodology and Evaluation Plan

1.8.1 Research Methodology

The primary objective of this research is to design and develop a model that enables **CIS** to handle high-resolution images. Initially, the research will focus on finding a method to map high-resolution images to low-resolution images. This phase will involve exploring various image processing, transformation, and machine learning-based techniques.

Furthermore, the research will aim to identify a suitable **GAN** model for generating low-resolution stego images. This step will include reviewing existing **GAN** architectures, particularly those used in image generation and transformation tasks. The performance, complexity, and applicability of different **GAN** models will be assessed to identify the best fit for the specific needs of this research. A **GAN** model that demonstrates high potential for generating high-quality low-resolution stego images will be selected.

Based on the findings, the next step will be the implementation phase. During this phase, the **CIS** model will be developed to generate low-resolution stego images from high-resolution secret images. Once the development phase is completed, the model will be trained using publicly available datasets. Following the conclusion of the training phase, the model will be evaluated to determine its effectiveness and performance. High level diagram of the methodology is show in the Figure **1.3**.

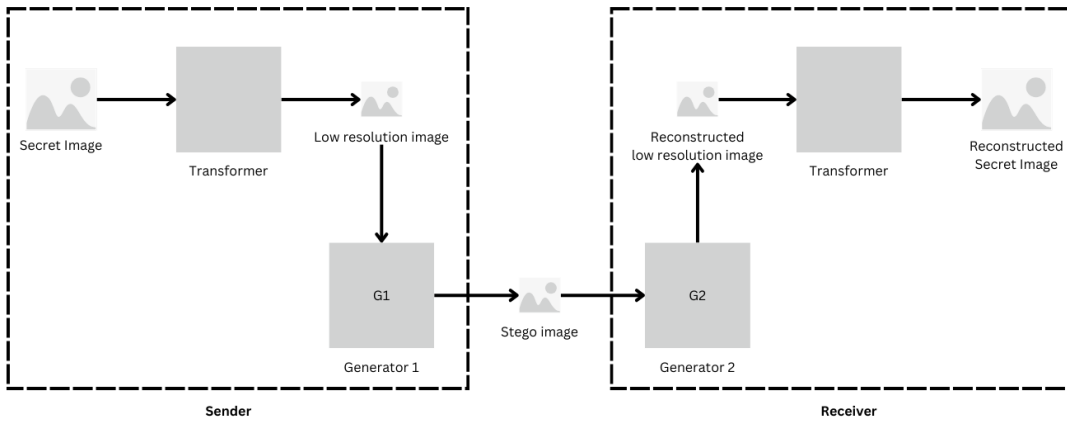


Figure 1.3: Methodology in high level

1.8.2 Evaluation Plan

1. Evaluation of the obtained stego image using existing steganalysis tools.

There are many steganalysis tools developed that are capable of detecting steganography. Therefore, to evaluate the outcome of the model built, we must subject the stego-images produced by the model to these existing steganalysis tools and determine the model's resistance to detection.

2. Evaluation of the reconstructed secret image.

To evaluate the quality of the reconstructed secret image, several methods will be employed. Quantitative metrics like Peak Signal-to-Noise Ratio (**PSNR**) and Mean Squared Error (**MSE**) will measure the difference between the original and reconstructed images, with higher **PSNR** and lower **MSE** values indicating better quality. The Structural Similarity Index (**SSIM**) will assess the perceived quality by comparing luminance, contrast, and structure, with values closer to 1 indicating better similarity.

3. Evaluation of the proposed model by comparison to the existing **CIS** models.

During this stage of evaluation, we will compare the newly developed model with existing models to assess whether the implementation has enhanced the capabilities of such models. This comparative analysis will help us understand the strengths and weaknesses of the new model and identify areas where improvements have been made, contributing to the advancement of the field of steganography.

1.9 Project Timeline

The gantt chart to illustrate the timeline of this research project is shown in Figure 1.4.

Chapter 02

2.1 Literature Review

This chapter details the comprehensive review undertaken on the existing literature relevant to this study. [2.1.1](#) includes an overview of theories utilized to build steganographic methods. [2.1.2](#) details an introduction to steganography and an overview of the methods that have been employed in prior research related. Section 2.3 provides insight to image steganography and the respective approaches that have been adopted. Section 2.4 presents an overview of steganalysis and the various methods utilized. Finally, section 2.5 details the state-of-the-art related to GAN-based steganography. Moreover, this chapter highlights the limitations of the existing methods and identifies the research gap that this study seeks to address.

2.1.1 Theories

2.1.1.1 Convolutional Neural Networks

A [CNN](#) is a specialized type of deep neural network primarily designed for processing and analyzing visual data. [CNNs](#) are widely applied in various computer vision tasks such as *image classification*, *object detection*, and *semantic segmentation*, and they are increasingly used in other domains like *medical diagnosis*, *autonomous driving*, and *robotics*. Inspired by the human visual system, [CNNs](#) process data through a hierarchical structure, gradually extracting more abstract features from raw inputs.

At the core of a [CNN](#) is the *convolutional operation*, which involves applying a set of learnable filters (or *kernels*) that slide across the spatial dimensions of the input image. These filters are designed to detect local patterns such as edges, textures, or shapes. The output of this operation is a collection of *feature maps* that highlight the presence of specific features at different locations in the input.

A standard [CNN](#) architecture includes several types of layers:

- **Convolutional Layers:** These layers perform the convolution operation to

extract local features. Each filter produces a separate feature map by responding to specific patterns in the input.

- **Activation Functions:** To introduce non-linearity into the model, the output of the convolution is passed through a non-linear activation function, typically the Rectified Linear Unit (**ReLU**), which helps the network learn complex mappings.
- **Pooling Layers:** Also known as subsampling or downsampling layers, pooling reduces the spatial dimensions of the feature maps while retaining the most important information. The two common types are:
 - *Max Pooling*, which selects the maximum value within a sliding window.
 - *Average Pooling*, which computes the average of the values within the window.
- **Fully Connected Layers:** These layers come at the end of the network, where the extracted features are flattened and passed to one or more dense layers to perform classification or regression.

The training of **CNNs** involves minimizing a loss function using the backpropagation algorithm, which adjusts the network's parameters to reduce prediction errors. Optimization algorithms such as Stochastic Gradient Descent (**SGD**) or more advanced optimizers like Adam are commonly used to improve convergence during training.

CNNs have achieved remarkable performance on benchmark datasets, particularly in the ImageNet Large Scale Visual Recognition Challenge (**ILSVRC**), where they have consistently delivered state-of-the-art results and, in some cases, surpassed human-level performance.

Figure 2.1 illustrates the typical architecture of a **CNN**, highlighting the flow of data through convolutional layers, activation functions, pooling operations, and fully connected layers.

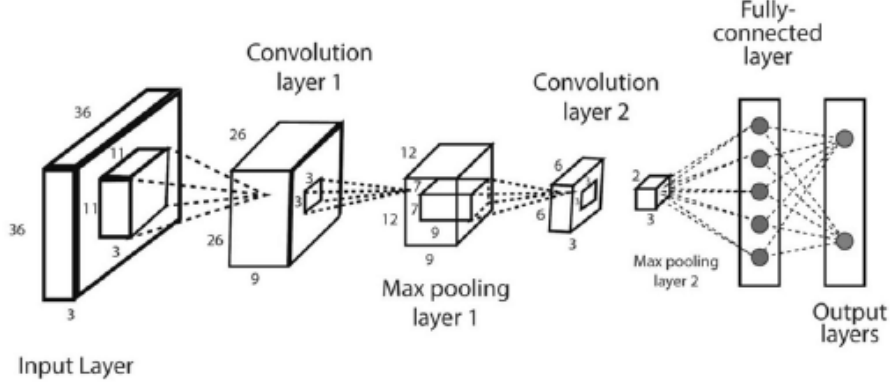


Figure 2.1: General architecture of a Convolutional Neural Network (CNN) (Ayub 2022).

2.1.1.2 Generative Adversarial Networks

The advent of GAN (Goodfellow et al. 2014) has led to significant advancements in steganographic methods over the years. GANs possess the unique capability to generate realistic, previously unseen images that closely mimic real-world images within a specific domain.

A GAN consists of two primary sub-models: the generator and the discriminator. During the training phase, these models engage in a min-max game, where the generator aims to create new image samples that closely resemble the training data, while the discriminator works to classify images as either real or fake. The generator produces an image that is an approximation of the input data, and the discriminator assesses whether this generated image can be distinguished from real data. The objective function for this process, as introduced by Goodfellow et al. (2014), is defined as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

This interplay between the generator and discriminator enables GANs to create high-quality, realistic images that are valuable for enhancing steganographic approaches. The architecture design of GANs is illustrated in Figure 2.2.

The generative capabilities of GANs inspired many steganographic models and these models have proved to improve the limitations existing within most traditional steganographic models. The key challenge that most traditional steganographic models face is the limitations of achieving high hiding capacity while ensuring high

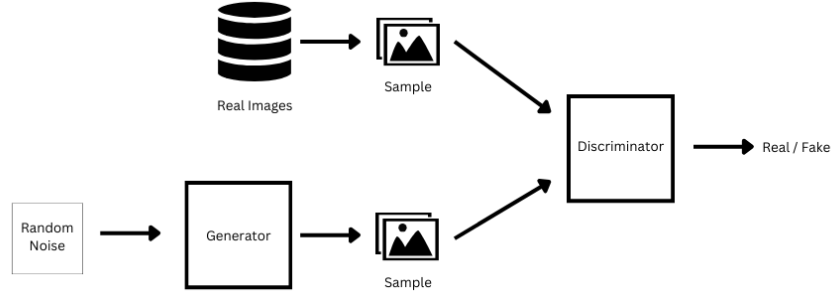


Figure 2.2: Architecture of a Generative Adversarial Network

security. Within this section, we will explore steganographic models which utilize GANs for information hiding and how these models have improved upon the existing limitations stated above.

2.1.2 Steganography

2.1.2.1 Overview

The study of steganography within the specific literature could be traced to the origins of the well known "Prisoners' Problem" (Simmons 1984) brought forward in 1983. In this Prisoners' problem, Alice and Bob, the two prisoners planning to escape from the prison. Both are allowed to communicate, however all their communications are inspected and passed through a warden, Eve. Eve will attempt to find any hidden communication between Alice and Bob, if there is any suspicion of information exchange between the two jailers, she will cut off the communication. In this scenario, Alice and Bob are trying to hide their secret communication, where steganography is depicted. Steganalysis refers to the inspection conducted by the warden, Eve, with the intention of detecting the secret information. Starting from the Prisoners' problem up until now, researchers have implemented numerous steganography and steganalysis methods for information hiding and detecting.

Until now, various communication mediums such as text, image, audio, and video have been used to implement steganography methods. Our main intention was to focus on image steganography. As I discussed earlier traditional image steganography methods modify the cover image to embed the secret data. With the advancement of deep learning, steganalysis tools have become more effective. Steganalysis models like Ye'Net (Ye et al. 2017) and SRNet (Boroumand et al. 2019b) have high detection accuracy. Hence, the traditional image steganography methods could be easily detected. To address this issue researchers have proposed the novel steganography method called coverless image steganography (Zhou et al. 2015).

When considering the current implementation methods of image steganography, three primary methods can be identified:

1. Traditional steganography methods.
2. CNN based methods.
3. GAN based methods.

Traditional methods are frameworks which use methods that are not related to machine learning or deep learning algorithms. Many traditional methods are based on the LSB technique (Neeta, Snehel & Capacitiesesess 2006). With the advancement of deep learning, researchers increasingly apply deep learning methodologies to steganography. Two such methods are CNN and GAN based methods. CNN-based (Das et al. 2021b) approaches use deep convolutional neural networks for embedding and extracting secret images, while GAN-based methods employ various GAN variants, including Deep Convolutional Generative Adversarial Networks (DCGAN), Wasserstein Generative Adversarial Network (WGAN), CycleGAN, StarGAN, etc.

When focusing on GAN-based models, which will be our primary emphasis, they employ various strategies in data hiding, including:

1. Cover Modification.
2. Cover Selection.
3. Cover Synthesis.

2.1.2.2 Cover Modification-based Steganography

Cover modification based steganography tries to embed the secret data by modifying the cover image. Volkhonskiy et al. (2020) introduced the idea of using a GAN for steganography, presenting a model called Steganographic GAN (SGAN). This model includes three networks: a Generator, a Discriminator, and a Steganalysis classifier based on a DCGAN. The generator creates a cover image, after which a traditional method like LSB is applied for steganography. Once the embedding is done, the stego-image is evaluated by the steganalysis classifier to check for hidden information. Similarly, Shi et al. (2017) enhanced this concept with the Secure Steganographic GAN (SSGAN), utilizing a WGAN to generate the cover image. Another distinct approach was proposed by Wang et al. (2018), where the discriminator's role is to identify if the image has undergone steganography. Over the

years, various **GAN** applications have been employed to develop image steganography models. However, these methods struggle against advanced steganalysis tools, as any embedding operation leaves detectable traces within the cover image.

2.1.2.3 Cover Selection-based Steganography

In this case No embeddings were done to the cover image in this case which is similar to image retrieval. Here, we build an image database and compute the hash sequence for each image using a robust hashing algorithm. Images sharing identical hash sequences with the corresponding segments of the confidential message are selected as the carrier images. While this method exhibits resistance to steganalysis, it faces the challenge of having a very limited payload, so that it's impractical for real-world applications. **Zhou et al. (2015)** proposed a such method where image database is first constructed by collecting number of images. Then, for each image in the database, its hash sequence is generated by a robust hashing algorithm. Afterward, all of these images are indexed according to their hash sequences to build an inverted index structure. **Li et al. (2024)** proposed a novel algorithm leverages coverless image datasets (CIDs) and a mapping rule to establish correspondence between hash sequences and images. Secret information is embedded in images with matching hash sequences. At the receiver side, robust Speeded-Up Robust Features (**SURF**) features are used to retrieve the hidden secrets. **Chen et al. (2022)** introduced an innovative **CIS** method that leverages both image selection and StarGAN.

2.1.2.4 Cover Synthesis-based Steganography

Due to limitations in cover modification-based and cover selection-based approaches novel method based on cover synthesis-based image steganography has been introduced which is also known as coverless image steganography. This involves generating a stego image, which is an image containing the embedded secret data. **Duan & Song (2018)** proposed a novel **CIS** method. In this approach, when a secret image is input into the generative model, an entirely different normal image is produced using a **WGAN**. **Hu et al. (2018)** proposed another **CIS** method where stego image is generated by feeding the established relationship between secret message and noise vector as input to the **DCGAN**. **Cao et al. (2020)** introduced a novel coverless steganography technique centered on the creation of anime characters. In this method, secret data is transformed into an attribute label of anime characters, which is then employed to generate these characters using **GAN**. A method similar to **Cao et al. (2020)** was introduced by **Chen et al. (2022)**, presenting an original **CIS** approach. In this method, secret information is converted into a sequence of face attributes, which is

then used to generate a face image by adjusting the original face attributes based on the sequence derived from the secret message, utilizing a StarGAN.

One major drawback in CIS is maintaining high reconstruction quality of the secret image. To address this issue Liu, Ma, Guo, Hou, Schaefer, Wang, Wang & Fang (2020) proposed a method called "Coverless Image Steganography Using Cam-GAN" to address the challenges of coverless image. This method capable of hiding full image-to-image while maintaining imperceptibility and high reconstruction quality in the secret image. Also, Li et al. (2021) brought forward a content consistency CIS method which extracts content information of the secret image and encodes it into the carrier image. This approach retains the original details of the secret image when reconstructed from the extraction.

2.1.3 Steganalysis

Steganalysis, considered the counterpart to steganography, aims to detect the presence of hidden information. Analyzing existing steganalysis models is essential for understanding how steganographic content is identified, enabling the development of more robust steganographic techniques and selecting the most suitable steganalysis model to assess the security of multi-image steganographic systems. This section discusses the fundamental concepts of steganalysis and explores state-of-the-art models.

Steganalysis methods can be categorized into three main types: passive, active, and malicious steganalysis, which can be illustrated through the Prisoner's Problem analogy described in subsection 2.1.1.

Passive Steganalysis: In this approach, the warden (Eve) acts as a passive observer whose sole objective is to detect the presence of hidden information. If Eve identifies a secret message, her only action is to prevent its transmission (Anderson & Petitcolas 1998). A passive warden lacks the ability to modify or destroy the message and can only authorize or block its delivery.

Active Steganalysis: An active warden, represented by Eve, not only aims to detect the presence of hidden information but also actively modifies the image to remove the embedded content. This altered image is then sent to the intended recipient. By altering the image, the active warden ensures that secret information is destroyed or disrupted, preventing successful communication.

Malicious Steganalysis: In this scenario, the warden not only detects the presence of a hidden message but also attempts to identify the steganographic method used to embed the secret and may generate new cover images to impersonate the sender, thereby misleading the recipient (Francia & Gomez 2006).

Initial steganalysis techniques were developed to detect steganographic methods based on **LSB** modifications. These methods involve converting the stego-image into its binary format and examining the LSBs for inconsistencies that reveal the presence of hidden data.

Westfeld & Pfitzmann (2000) proposed the first statistical steganalysis model based on identifying Pairs of Values (**POVs**) exchanged during message embedding. StegExpose (**Boehm 2014**) is a publicly available steganalysis toolkit which has proven attacks in a timeefficient manner.

Subsequently, steganalysis methods were built based on various deep learning models. **Qian et al. (2015)** put forward the GNCNN model, the first convolutional neural network **CNN** based model to detect the existence of steganographic content.

Xu et al. (2016) proposed the model Xu-Net which has proved to yield satisfactory results in the field of steganalysis. This model was considered to be a noteworthy model as it was the first network to achieve comparable results to the state-of-the-art steganographic methods based on the two-step learning approach. Xu-Net accepts gray-scale images of size 512×512 as input and produces a vector containing two probabilities as an output. Each of these probability values represent how likely the given image can be categorized into a stego-image class or normal image class.

In 2017, the first model that has the capabilities to surpass existing methods on the tiny BOSS database called Ye-Net (**Ye et al. 2017**) was proposed by Ye et al. Ye-Net is created to take gray-scale images of size 256×256 as input and gives its classification as output.

Modern steganography models such as **Fu et al. (2020)**, **Liu, Ma, Guo, Hou, Schaefer, Wang, Wang & Fang (2020)**, **Cao et al. (2020)** use one or more of the steganalysis tools such as StegExpose, GNCNN and Xu-Net in order to test the security of the developed method.

2.1.4 Research Gap

The increasing demand for information hiding has led to significant advancements in steganography, particularly in the domain of **CIS**. However, existing **CIS** methods typically generate stego images with the same resolution as the original secret images, which poses challenges related to high storage capacity and bandwidth requirements. These limitations hinder the practical application of steganography in environments with restricted or costly storage and bandwidth.

Moreover, current approaches fail to address the critical need for generating low-resolution stego images from high-resolution secret images. Such a capability

is essential to optimize storage and bandwidth utilization, as well as to improve resistance to steganalysis tools that aim to detect hidden information. Additionally, while GANs have demonstrated promise in generating high-quality, realistic images that closely resemble natural images, their integration into **CIS** for generating low-resolution stego images remains underexplored.

This research aims to fill this gap by proposing the use of Generative Adversarial Networks (**GANs**) to develop a novel coverless image steganography method that generates low-resolution stego images. This approach seeks to improve storage efficiency, optimize network transmission, enhance resistance to steganalysis, and ensure the integrity and security of the hidden data.

Chapter 03

3.1 Research Design

The research design followed in this research consists of 10 main phases; Defining of Research Questions, Literature review, Data collection, High-Resolution Restoration Model , High-Resolution Restoration Model Evaluation, Stego Image Generation Model, Stego Image Generation Model Evaluation, Model Integration, Integrated Model Evaluation and Conclusion which will be elaborated further within this section. Figure [3.1](#) illustrates the high level flow of these research phases along with their respective outcomes.

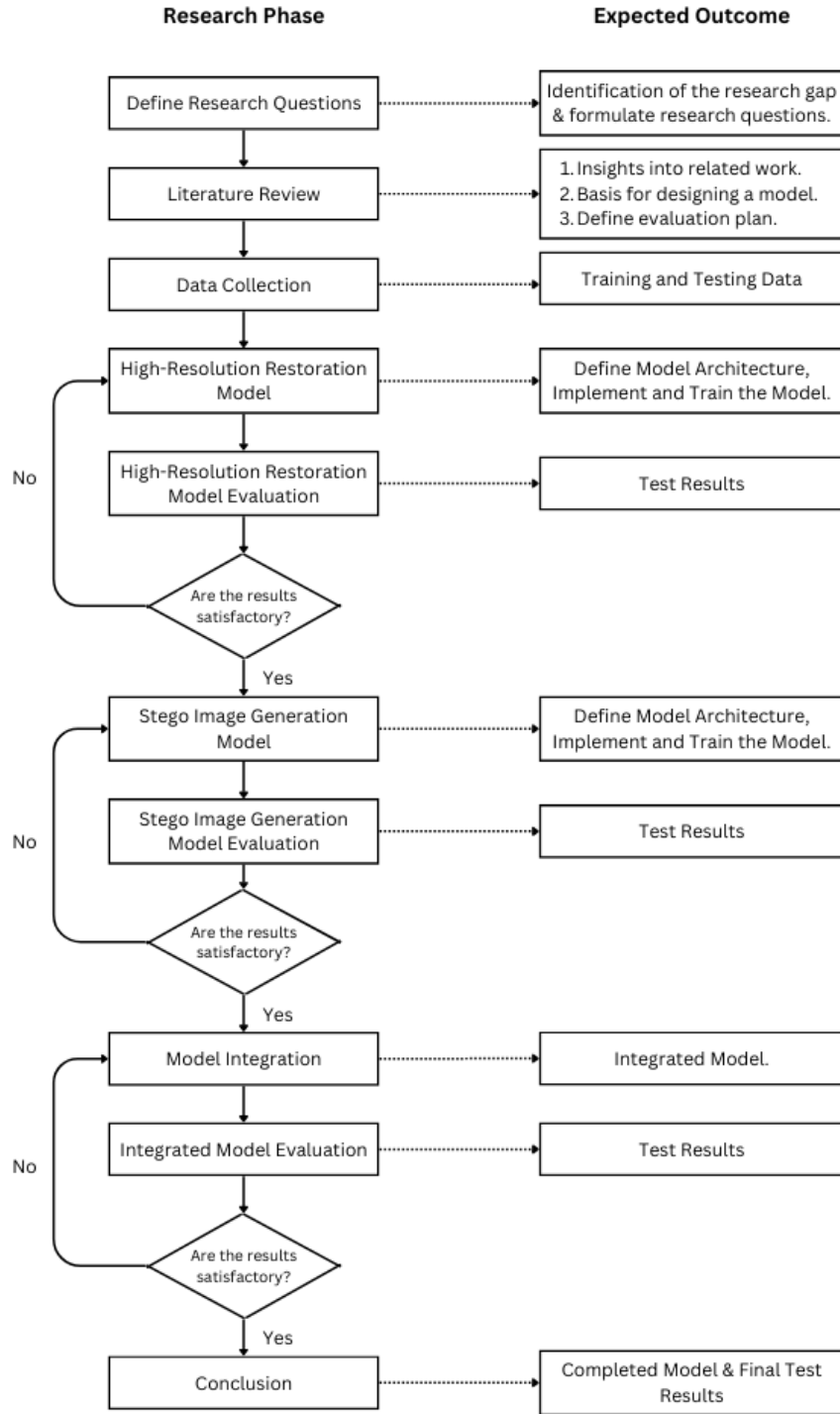


Figure 3.1: Research Design

3.1.1 Step 1: Defining of Research Questions

During this phase we identified the research gap as stated in subsection 1.4 and the relevant research questions were formulated as follows:

1. How can Generative Adversarial Networks (GAN) be utilized to develop a novel coverless image steganography method that generates low-resolution stego images from high-resolution secret images, while maintaining data integrity and security?
2. What are the most effective Generative Adversarial Network (GAN) architectures and techniques for embedding data in low-resolution stego images to maximize data capacity and minimize perceptual distortion?
3. How does the reduction in resolution of stego images impact the detectability of hidden data by standard steganalysis tools?

3.1.2 Step 2: Literature Review

A thorough literature review was conducted as an initial phase of this project to identify novel approaches that researchers have utilized to achieve image steganography. By doing so we identified the strengths and weaknesses of each of these research approaches. The conclusion of this investigation allowed us to obtain a basis for the architecture design of the two models and facilitate the formulation of the evaluation plan.

3.1.3 Step 3: Data Collection

In order to train and evaluate the proposed high to low resolution steganography model, we utilized multiple publicly available datasets, each serving distinct roles in the steganographic pipeline:

- **Cover Image Synthesis:** For the generation of stego images, we employed the **DELAUNAY** (Gontier et al. 2022) dataset, which contains stylized artistic and abstract images. All cover images were resized to a fixed resolution of 256×256 pixels to serve as low-resolution stego targets. A total of 1350 images from DELAUNAY were used for cover image generation during training.
- **Secret Images:** High-resolution secret images were selected from the **Flickr2K** (Lim et al. 2017) and **Div2K** (Timofte et al. 2017) datasets. These original high-resolution images were resized to 512×512 pixels to ensure consistency across

training and evaluation phases. A total of 1500 images were combined from both datasets for training, while an independent test set of 500 high-resolution images was used for evaluation purposes.

- **Experimental Dataset:** For preliminary experimentation and model prototyping, we utilized the **summer2winter Yosemite** dataset (Zhu et al. 2017), available at <https://www.kaggle.com/datasets/suyashdamle/cyclegan>. This dataset comprises 256×256 resolution images. Its manageable size and resolution made it particularly suitable for rapid experimental iterations given time and computational resource constraints.

This multi-dataset approach allowed for comprehensive training, robust evaluation, and efficient experimentation, ensuring the proposed model was rigorously validated under various practical scenarios.

3.1.4 Step 4: High-Resolution Restoration Model

In this work, we focus on generating a low-resolution image from a given high-resolution input image. Subsequently, the same high-resolution image is reconstructed using the generated low-resolution image as input. This process demonstrates the capability to compress and encode high-resolution image data into a lower-resolution format, which can then be effectively utilized to recover the original high-resolution image. This approach is intended to optimize storage and transmission efficiency while preserving the integrity of the reconstructed image. This process is shown in Figure 3.2.

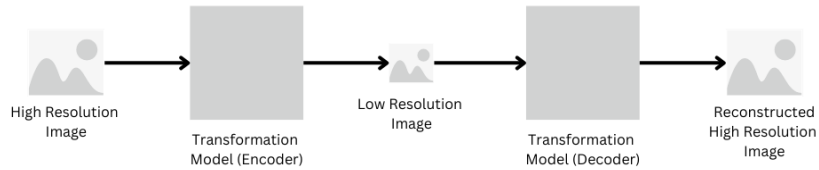


Figure 3.2: Overview of High-Resolution Restoration Model

To address this task, we have developed two distinct models. The first model is based on an autoencoder architecture, designed to efficiently learn the mapping between high-resolution and low-resolution images. The second model combines bicubic interpolation (Fadnavis 2014) with a refinement network, aimed at enhancing the quality of the low-resolution image and improving the accuracy of the high-resolution reconstruction. These models offer different approaches to the problem,

each leveraging unique techniques to optimize image compression, reconstruction, and overall performance, which we will be discussed in detail in later sections.

3.1.5 Step 5: High-Resolution Restoration Model Evaluation

The performance of the high-resolution restoration model will be evaluated using quantitative metrics such as [PSNR](#), [MSE](#), and [SSIM](#). These metrics will be used to assess the quality of the reconstructed high-resolution images and compare the performance of the two models. The evaluation results will provide insights into the effectiveness of the models in preserving image quality and integrity during the compression and reconstruction process.

3.1.6 Step 6: Stego Image Generation Model

In this phase, the stego image is generated using the low-resolution image produced by the encoder of the High-Resolution Restoration Model. This is achieved through the application of Generator 1, which creates the stego image. Subsequently, Generator 2 is employed to reconstruct the original low-resolution image from the stego image. This dual-generator framework ensures that the generated stego image can be effectively utilized for secure transmission while maintaining the ability to accurately reconstruct the initial low-resolution image. This process is shown in Figure [3.3](#).



Figure 3.3: Overview of Stego Image Generation Model

To address this task, we are planned to use CycleGAN ([Zhu et al. 2017](#)) architecture, which is a generative adversarial network designed for image-to-image translation tasks. This architecture is well-suited for the stego image generation process, as it enables the generation of stego images from low-resolution input images while preserving the integrity and security of the embedded data.

3.1.7 Step 7: Stego Image Generation Model Evaluation

This evaluation also same as the evaluation of the High-Resolution Restoration Model. The performance of the stego image generation model will be evaluated using quantitative metrics such as **PSNR**, **MSE**, and **SSIM**. These metrics will be used to assess the of the original low-resolution image and the quality of the reconstructed low-resolution image from the stego image. The evaluation results will provide insights into the effectiveness of the model in generating stego images and preserving the integrity of the embedded data.

3.1.8 Step 8: Model Integration

In this phase, the High-Resolution Restoration Model and the Stego Image Generation Model are integrated to create a comprehensive framework for Low-Resolution Coverless Image Steganography for High-Resolution Images, utilizing **GANs**. This integrated approach leverages the strengths of both models to ensure efficient generation of low resolution stego image and reconstruction of high-resolution image, enabling secure and effective image steganography. Overall architecture of the integrated model is shown in Figure 3.4.

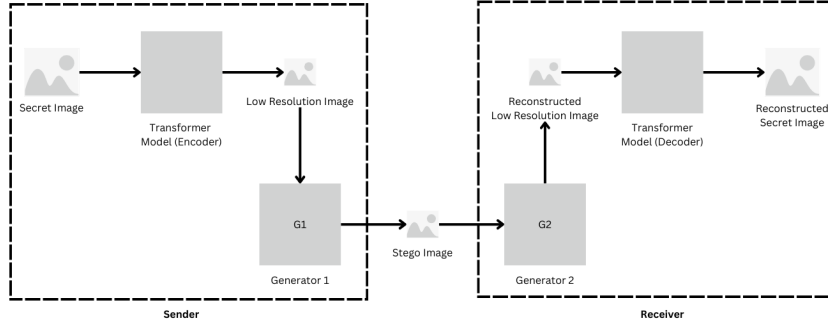


Figure 3.4: Overview of Integrated Model

3.1.9 Step 9: Integrated Model Evaluation

The integrated model implemented from the previous stage will be evaluated according to the evaluation plan defined in subsection 3.3. The conclusion of this phase will result in test results reflecting the performance of the model. If the results obtained from this phase are proven to accomplish the research aim we will conclude the project. However, if the results obtained are not satisfactory we will either redesign the model or adjust hyper-parameters accordingly and iterate through the process until we obtain the required results.

Chapter 04

4.1 Model Architectures and Experimental Results

This section presents a comprehensive overview of the experiments conducted throughout the research. The experimental process was systematically categorized into three main stages. The first category focuses on the transformation and reconstruction of high-resolution images into low-resolution representations, evaluating the model’s capability to preserve essential visual features during resolution reduction. The second category encompasses the generation of stego images, where the objective was to embed secret information within low-resolution images while maintaining imperceptibility and data integrity. The final category combines the findings of the previous two to develop and evaluate a novel approach for low-resolution coverless image steganography of high-resolution images. Each experiment is discussed in detail, including the underlying model architectures, training configurations, and performance evaluation metrics used to assess the effectiveness of the proposed methods.

4.1.1 High-Resolution Restoration Model Experiments

This subsection details the experiments conducted to explore various methodologies for restoring high-resolution images from their low-resolution counterparts. The primary objective was to assess different restoration techniques in terms of their ability to reconstruct high-quality images while preserving semantic and structural fidelity.

Three distinct experiments were carried out under this category. **Experiment 1** investigates traditional *interpolation techniques*—including nearest-neighbor, bilinear, and bicubic interpolation—as baseline methods for high-resolution image restoration. **Experiment 2** explores a *deep learning-based approach using auto-encoders*, evaluating their effectiveness in learning compressed representations and reconstructing high-resolution outputs with improved detail retention. **Experiment**

3 introduces a *hybrid method* that combines *bicubic interpolation with a refinement network*, aiming to enhance the initial interpolated output through a trainable model that corrects artifacts and sharpens features.

All experiments detailed in this section were conducted using the summer2winter Yosemite dataset (Zhu et al. 2017), with an initial image resolution of 256×256 , which was downsampled to 128×128 . After determining the optimal approach for High-Resolution Restoration, we plan to extend the experiments to include high-resolution images of up to 512×512 .

4.1.1.1 Experiment 1: High-Resolution Image Restoration Using Interpolation Techniques.

Initially, we explored various interpolation techniques, including bilinear, bicubic, and nearest neighbor interpolation (Fadnavis 2014). These methods were analyzed to assess their effectiveness in downscaling and subsequently upscaling image resolutions, serving as a foundation for high-resolution image restoration.

Bilinear Interpolation

Bilinear interpolation estimates values at arbitrary positions by calculating the weighted average of the four nearest pixels to the specified input coordinates and assigning the resulting value to the output coordinates. This process involves performing two linear interpolations along one axis, followed by a linear interpolation in the perpendicular direction (Fadnavis 2014). The interpolation kernel can be expressed as:

$$u(x) = \begin{cases} 0, & \text{if } |x| > 1 \\ 1 - |x|, & \text{if } |x| < 1 \end{cases}$$

where ‘x’ represents the distance between the two points being interpolated.

The experimental results were evaluated using quantitative metrics, including PSNR, MSE, and SSIM. Results were shown in Figure 4.1 and Table 4.1.

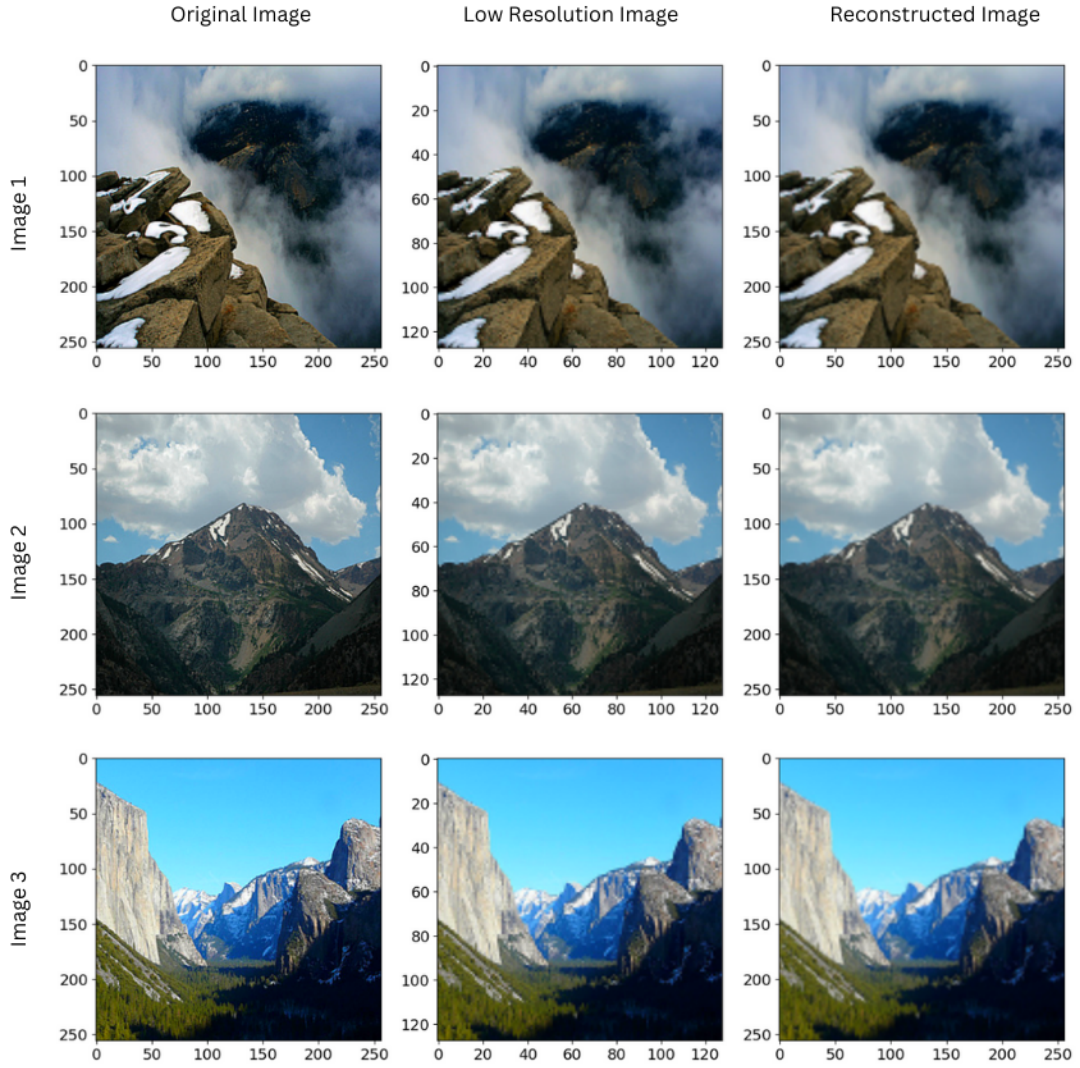


Figure 4.1: Bilinear Interpolation Results

Metric	Image 1	Image 2	Image 3
SSIM	0.73252	0.69140	0.70321
PSNR	71.2813	72.5612	70.6376
MSE	0.00484	0.00360	0.00561

Table 4.1: SSIM, PSNR, and MSE values for Images

Bicubic Interpolation

Bicubic interpolation is an enhancement over cubic interpolation applied to a two-dimensional regular grid. It produces a smoother interpolated surface compared

to the surfaces generated by bilinear and nearest-neighbor interpolation methods. Bicubic interpolation employs polynomials or the cubic convolution algorithm to estimate values at arbitrary positions.

In cubic convolution interpolation, the grey-level value at a specified input coordinate is determined by the weighted average of the 16 closest pixels, considering a 4×4 grid. For bicubic interpolation (which is cubic convolution applied in two dimensions), the interpolation function requires 16 grid points—eight grid points (two on either side) along both the horizontal and vertical axes. The obtained results were shown in Figure 4.2 and Table 4.2.

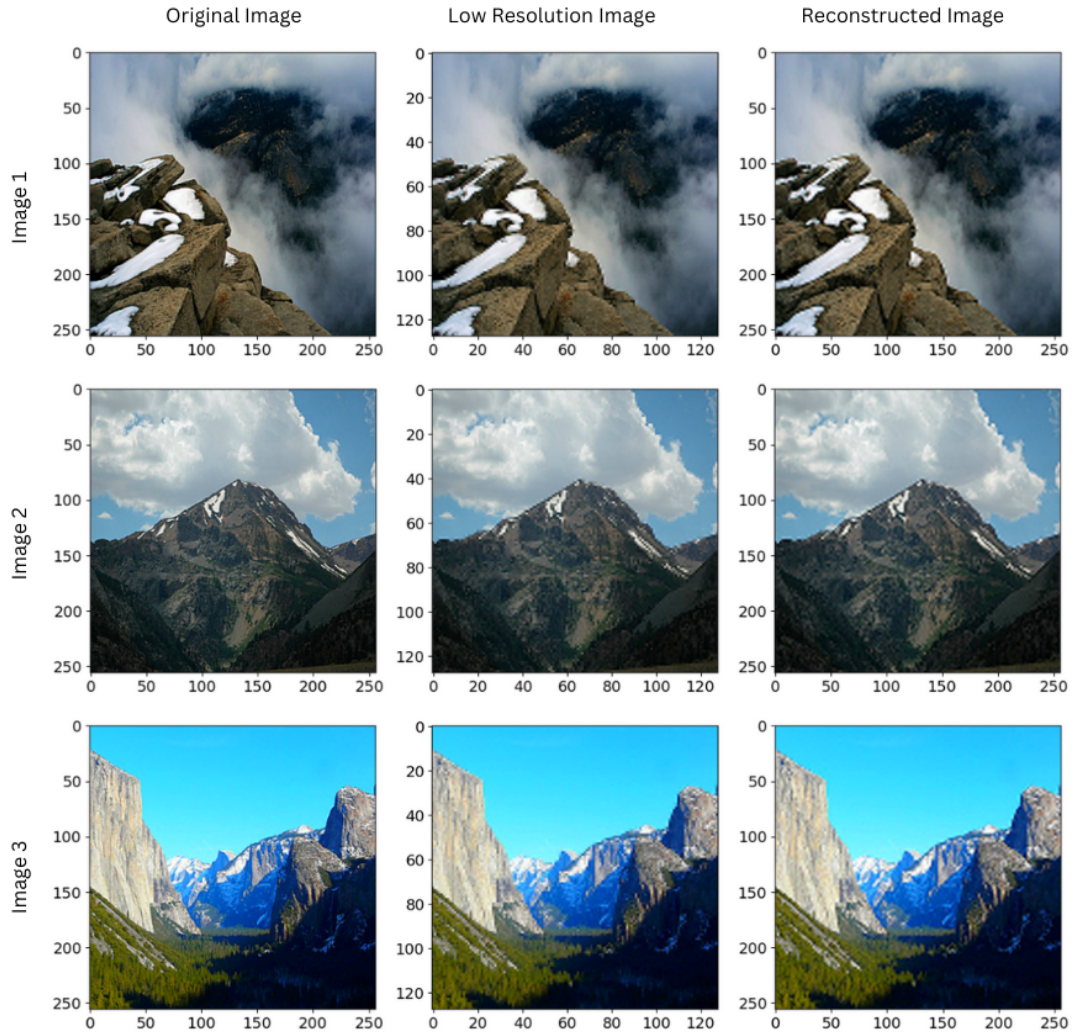


Figure 4.2: Bicubic Interpolation Results

Metric	Image 1	Image 2	Image 3
SSIM	0.78381	0.72648	0.75241
PSNR	71.9927	72.6402	70.8268
MSE	0.00410	0.00354	0.00538

Table 4.2: SSIM, PSNR, and MSE values for Images

Nearest Neighbor Interpolation

Nearest neighbor interpolation is the simplest form of interpolation. In this method, each output pixel is assigned the value of the nearest sample point from the input image. The interpolation kernel for nearest neighbor interpolation is defined as:

$$h(x) = \begin{cases} 1, & \text{if } |x| < 0 \\ 0, & \text{if } |x| > 0 \end{cases}$$

The frequency response of the nearest neighbor kernel is given by:

$$H(\omega) = \text{sinc}\left(\frac{\omega}{2}\right) \quad (4.1)$$

Although this method is computationally efficient, the resulting image quality is generally poor. This is because the Fourier transform of a rectangular function is equivalent to a sinc function, which has a rapidly decaying passband. Additionally, the sinc function exhibits prominent side lobes when analyzed on a logarithmic scale, leading to visible artifacts and a loss of smoothness in the interpolated image. The obtained results were shown in Figure [4.3](#) and Table [4.3](#).

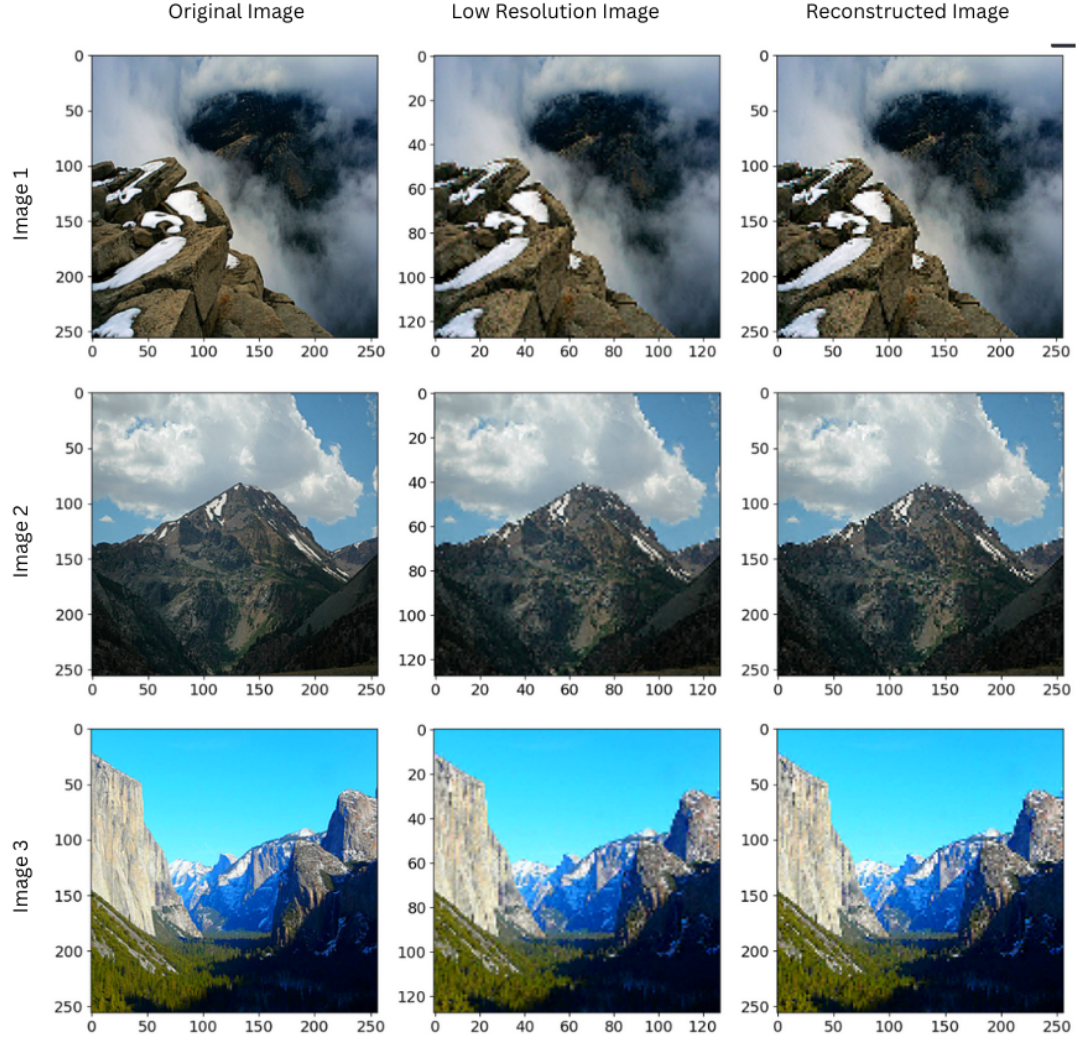


Figure 4.3: Nearest Neighbor Interpolation Results

Metric	Image 1	Image 2	Image 3
SSIM	0.66818	0.61571	0.66319
PSNR	68.1302	69.6594	68.0526
MSE	0.01000	0.00703	0.01018

Table 4.3: SSIM, PSNR, and MSE values for Images

The results of this experiment indicated that bicubic interpolation provided superior performance for high-resolution image restoration compared to other interpolation methods. This finding aligns with observations mentioned in the litera-

ture (Fadnavis 2014), where bicubic interpolation is noted for its ability to produce smoother and more visually appealing results.

Metric	Bilinear			Bicubic			Nearest Neighbour		
	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3
SSIM	0.73252	0.69140	0.70321	0.78381	0.72648	0.75241	0.66818	0.61571	0.66319
PSNR	71.2813	72.5612	70.6376	71.9927	72.6402	70.8268	68.1302	69.6594	68.0526
MSE	0.00484	0.00360	0.00561	0.00410	0.00354	0.00538	0.01000	0.00703	0.01018

Table 4.4: SSIM, PSNR, and MSE values for Images across Bilinear, Bicubic, and Nearest Neighbour Interpolation

4.1.1.2 Experiment 2: Auto-Encoder Based High-Resolution Image Restoration.

Building on the approach proposed by (Wang et al. (2016)), an Auto-Encoder was implemented from scratch for high-resolution image restoration. The model consists of two primary networks: the encoder network, which takes the high-resolution image as input and generates a low-resolution image, and the decoder network, which takes the low-resolution image and reconstructs the original high-resolution image.

Additionally, a novel loss function was introduced for training the autoencoder. The loss function is a weighted sum of three distinct losses: **MSE** loss, Structural **SSIM** loss (Brunet et al. 2007), and Perceptual loss (Johnson et al. 2016). These combined loss components enable the model to optimize both pixel-level accuracy and perceptual quality during training.

The architecture of the model, along with the details of the loss function, is presented below.

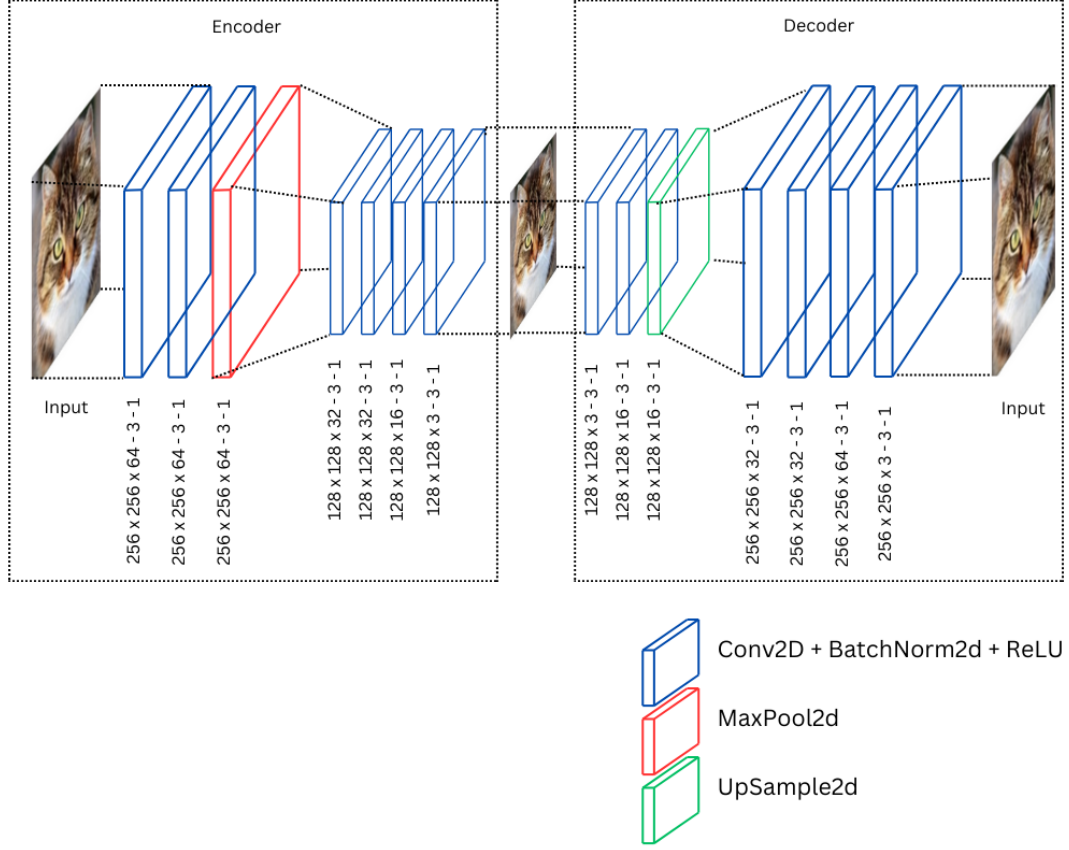


Figure 4.4: Auto-Encoder Architecture for High-Resolution Image Restoration

The loss function for the Auto-Encoder model is defined as:

$$L = \alpha \cdot \text{MSE}(I, \hat{I}) + \beta \cdot \text{SSIM}(I, \hat{I}) + \gamma \cdot \text{PerceptualLoss}(I, \hat{I})$$

where:

- I represents the original high-resolution image.
- \hat{I} represents the reconstructed high-resolution image.
- α , β , and γ are the weights assigned to the MSE, SSIM, and Perceptual loss components, respectively.
- $\text{MSE}(I, \hat{I})$ denotes the Mean Squared Error between the original and reconstructed images.
- $\text{SSIM}(I, \hat{I})$ represents the Structural Similarity Index between the original and reconstructed images.

- $\text{PerceptualLoss}(I, \hat{I})$ is the perceptual loss calculated using a pre-trained VGG-16 network.

Here the alpha, beta, and gamma values were fine-tuned using hyper parameter tuning to optimize the performance of the model. Details of the hyperparameter tuning process and the final values selected for alpha, beta, and gamma are provided in the Figure 4.5.

```
[I 2024-11-05 13:56:21,916] A new study created in memory with name: no-name-3f8d48d8-c446-47b0-b632-d75bd376b0bd
[I 2024-11-05 14:02:53,787] Trial 0 finished with value: 3.2302857745777476 and parameters: {'ratio_mse': 0.16908325123172743, 'ratio_ssim': 0.3872151639886324, 'ratio_perceptual': 0.8143030218342777}. Best is trial 0 with value: 3.2302857745777476.
[I 2024-11-05 14:09:16,662] Trial 1 finished with value: 1.3384872422073826 and parameters: {'ratio_mse': 0.17337526164073858, 'ratio_ssim': 0.39853952737751486, 'ratio_perceptual': 0.3822917687516665}. Best is trial 1 with value: 1.3384872422073826.
[I 2024-11-05 14:15:38,213] Trial 2 finished with value: 0.896605645165299 and parameters: {'ratio_mse': 0.1574968094699427, 'ratio_ssim': 0.551228446281267, 'ratio_perceptual': 0.2515251042714789}. Best is trial 2 with value: 0.896605645165299.
[I 2024-11-05 14:21:58,620] Trial 3 finished with value: 0.7121031320456302 and parameters: {'ratio_mse': 0.10598237865724754, 'ratio_ssim': 0.4089382069348212, 'ratio_perceptual': 0.22440512859141803}. Best is trial 3 with value: 0.7121031320456302.
[I 2024-11-05 14:28:18,349] Trial 4 finished with value: 0.6295770659591212 and parameters: {'ratio_mse': 0.12812405908007796, 'ratio_ssim': 0.5193316407509885, 'ratio_perceptual': 0.20478279033229008}. Best is trial 4 with value: 0.6295770659591212.
[I 2024-11-05 14:34:37,620] Trial 5 finished with value: 0.4789924847357201 and parameters: {'ratio_mse': 0.8395066986847585, 'ratio_ssim': 0.4146408975065028, 'ratio_perceptual': 0.146932266033763}. Best is trial 5 with value: 0.4789924847357201.
[I 2024-11-05 14:40:56,819] Trial 6 finished with value: 0.6562123569575223 and parameters: {'ratio_mse': 0.49713723991984693, 'ratio_ssim': 0.3868475866395422, 'ratio_perceptual': 0.23133114166745658}. Best is trial 5 with value: 0.4789924847357201.
[I 2024-11-05 14:47:18,469] Trial 7 finished with value: 1.8047183535315774 and parameters: {'ratio_mse': 0.8828171009877889, 'ratio_ssim': 0.5742378967651192, 'ratio_perceptual': 0.7017376028701387}. Best is trial 5 with value: 0.4789924847357201.
[I 2024-11-05 14:53:48,513] Trial 8 finished with value: 1.7789217631022136 and parameters: {'ratio_mse': 0.25688945695106673, 'ratio_ssim': 0.4126355116548107, 'ratio_perceptual': 0.7293542750543155}. Best is trial 5 with value: 0.4789924847357201.
[I 2024-11-05 15:00:17,321] Trial 9 finished with value: 1.5550692840055986 and parameters: {'ratio_mse': 0.8173182612785326, 'ratio_ssim': 0.809773082764789, 'ratio_perceptual': 0.56437201028417}. Best is trial 5 with value: 0.4789924847357201.

Best Parameters: {'ratio_mse': 0.8395066986847585, 'ratio_ssim': 0.4146408975065028, 'ratio_perceptual': 0.146932266033763}
```

Figure 4.5: Hyperparameter Tuning Process for the Loss Function

So the best values for alpha, beta, and gamma were found to be 0.8395066986847585, 0.4146408975065028, and 0.146932266033763, respectively.

The final network was trained using the summer2winter Yosemite dataset (Zhu et al. 2017), for 500 epochs, with a batch size of 8. And the results were evaluated using the same metrics as in the previous experiment. The results were shown in Figure 4.6 and Table 4.5.

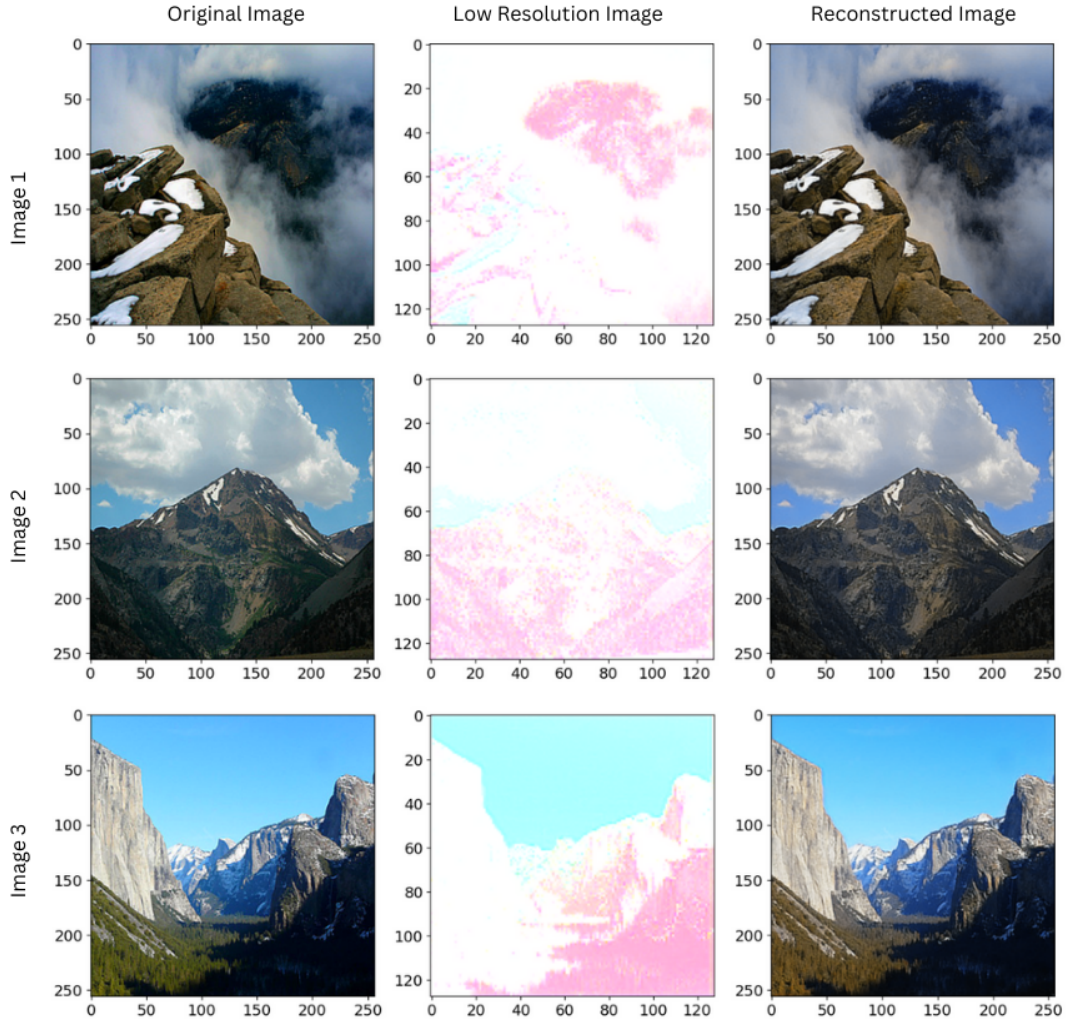


Figure 4.6: Auto-Encoder Results

Metric	Image 1	Image 2	Image 3
SSIM	0.88446	0.89952	0.84630
PSNR	77.3171	77.7219	75.1895
MSE	0.00669	0.00778	0.01338

Table 4.5: SSIM, PSNR, and MSE values for Images

4.1.1.3 Experiment 3: Bicubic Interpolation Enhanced with Refinement Network for High-Resolution Image Reconstruction.

Since bicubic interpolation demonstrated superior performance in the initial experiments in Section 4.1.1.1 we sought to enhance the bicubic interpolation method by incorporating a refinement network. The refinement network is designed to refine the reconstructed high-resolution image obtained from bicubic interpolation, thereby improving the overall quality of the image. The overall architecture of the model is shown in Figure 4.7.

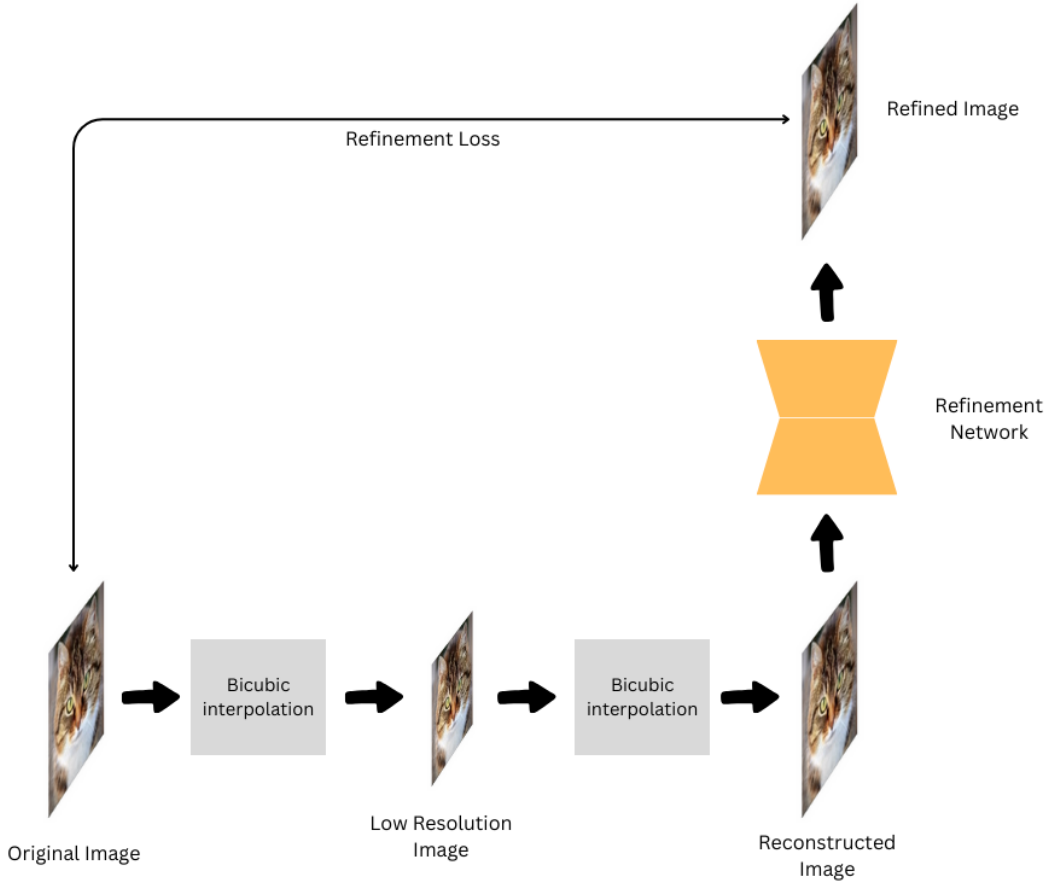


Figure 4.7: Bicubic Interpolation Enhanced with Refinement Network

The refinement network is adapted from CBDNet, proposed by Guo et al. (2019), which is a lightweight architecture originally designed for image denoising. For this experiment, the CBDNet structure was modified to serve as the refinement network, utilizing the novel loss function previously introduced for the Auto-Encoder in section 4.1.1.2. The objective of this adapted architecture is to enhance the qual-

ity of the low-resolution image and improve the overall accuracy of high-resolution reconstruction. The loss function for the refinement network is defined as follows:

$$L = \alpha \cdot \text{MSE}(I, \hat{I}) + \beta \cdot \text{SSIM}(I, \hat{I}) + \gamma \cdot \text{PerceptualLoss}(I, \hat{I})$$

where:

- I represents the original high-resolution image.
- \hat{I} represents the reconstructed high-resolution image.
- α , β , and γ are the weights assigned to the MSE, SSIM, and Perceptual loss components, respectively.
- $\text{MSE}(I, \hat{I})$ denotes the Mean Squared Error between the original and reconstructed images.
- $\text{SSIM}(I, \hat{I})$ represents the Structural Similarity Index between the original and reconstructed images.
- $\text{PerceptualLoss}(I, \hat{I})$ is the perceptual loss calculated using a pre-trained VGG-16 network.

The architecture of the refinement network, along with the details of the loss function, is presented below.

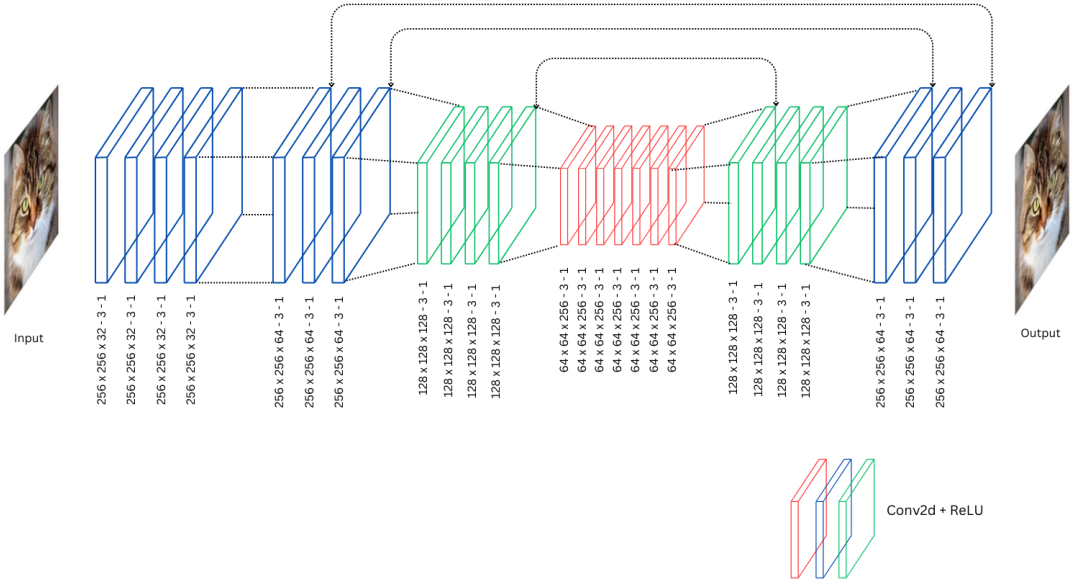


Figure 4.8: Refinement Network Architecture for High-Resolution Image Restoration

Here also the alpha, beta, and gamma values were fine-tuned using hyper parameter tuning to optimize the performance of the model. Details of the hyperparameter tuning process and the final values selected for alpha, beta, and gamma are provided in the Figure 4.9.

```
[I 2024-11-03 14:38:51,429] Trial 5 finished with value: 0.5845352736386386 and parameters: {'ratio_mse': 0.2643719337842054, 'ratio_ssim': 0.43165157803068255, 'ratio_perceptual': 0.1272735311878324}. Best is trial 5 with value: 0.5845352736386386.
[I 2024-11-03 15:01:44,703] Trial 6 finished with value: 2.747359478112423 and parameters: {'ratio_mse': 0.49523434080739803, 'ratio_ssim': 0.6436931239480057, 'ratio_perceptual': 0.7125486995234294}. Best is trial 5 with value: 0.5845352736386386.
[I 2024-11-03 15:25:08,216] Trial 7 finished with value: 1.9210493709101821 and parameters: {'ratio_mse': 0.7500293565012982, 'ratio_ssim': 0.4387871156969122, 'ratio_perceptual': 0.5021771250063702}. Best is trial 5 with value: 0.5845352736386386.
[I 2024-11-03 15:49:20,376] Trial 8 finished with value: 0.6484858267235033 and parameters: {'ratio_mse': 0.5366548941128672, 'ratio_ssim': 0.4658992098279364, 'ratio_perceptual': 0.13699234113518985}. Best is trial 5 with value: 0.5845352736386386.
[I 2024-11-03 16:16:13,415] Trial 9 finished with value: 3.1454803943634033 and parameters: {'ratio_mse': 0.3046433602818346, 'ratio_ssim': 0.5115450099147643, 'ratio_perceptual': 0.8510052042331564}. Best is trial 5 with value: 0.5845352736386386.

Best Parameters: {'ratio_mse': 0.2643719337842054, 'ratio_ssim': 0.43165157803068255, 'ratio_perceptual': 0.1272735311878324}
```

Figure 4.9: Hyperparameter Tuning Process for the Loss Function

So the best values for alpha, beta, and gamma were found to be 0.2643719337842054, 0.43165157803068255, and 0.1272735311878324, respectively.

The final network was trained using the summer2winter Yosemite dataset (Zhu et al. 2017), for 500 epochs, with a batch size of 8. And the results were evaluated using the same metrics as in the previous experiment. The results were shown in Figure 4.10 and Table 4.6.

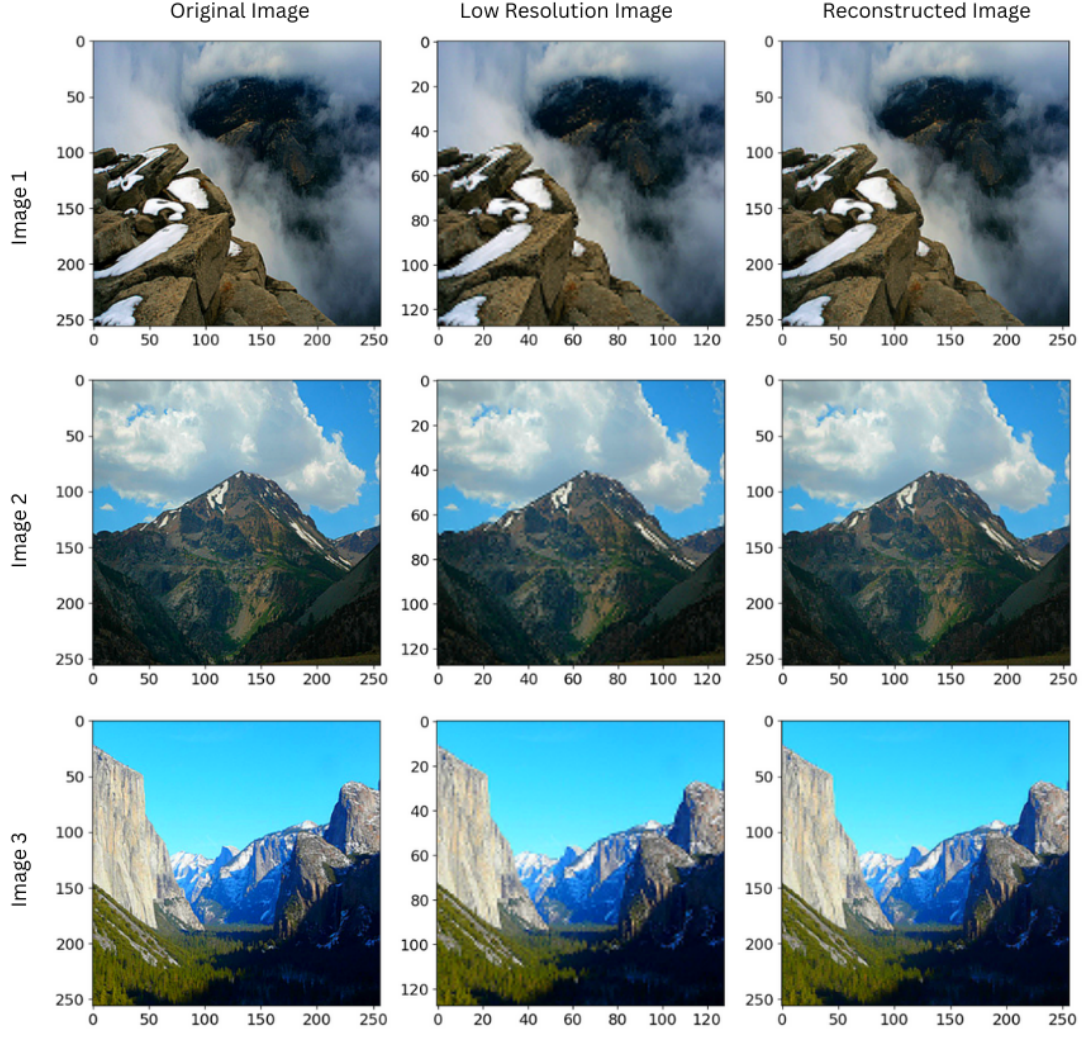


Figure 4.10: Bicubic Interpolation Enhanced with Refinement Network Results

Metric	Image 1	Image 2	Image 3
SSIM	0.92134	0.91472	0.92103
PSNR	77.3680	78.4892	76.5696
MSE	0.00512	0.00380	0.0594

Table 4.6: SSIM, PSNR, and MSE values for Images

A comparison of the results from the three experiments reveals that the bicubic interpolation method combined with the refinement network delivers superior performance in terms of **SSIM** and **PSNR** values. Conversely, the standalone bicubic

interpolation method yields the lowest **MSE**. Table 4.7 presents the comprehensive metrics for all three experiments. The findings demonstrate that the bicubic interpolation enhanced with the refinement network surpasses both the Auto-Encoder and standard bicubic interpolation methods, achieving higher image quality and improved restoration accuracy.

Metric	Bicubic Interpolation			Auto-Encoder			Bicubic with Refinement		
	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3	Image 1	Image 2	Image 3
SSIM	0.78381	0.72648	0.75241	0.88446	0.89952	0.84630	0.92134	0.91472	0.92103
PSNR	71.9927	72.6402	70.8268	77.3171	77.7219	75.1895	77.3680	78.4892	76.5696
MSE	0.00410	0.00354	0.00538	0.00669	0.00778	0.01338	0.00512	0.00380	0.00594

Table 4.7: Combined SSIM, PSNR, and MSE values for Images across different experiments: Bicubic Interpolation, Auto-Encoder, and Bicubic with Refinement Network

4.1.2 Stego Image Generation Model

This subsection outlines the experimental work conducted on stego image generation by modifying the CycleGAN training framework proposed by [Zhu et al. \(2017\)](#). The aim was to develop a model capable of embedding high-resolution secret images into low-resolution cover images in a coverless manner, while enabling the recovery of the original secret content.

To achieve this, the standard CycleGAN architecture was adapted by modifying its loss functions to better suit the objectives of steganography, particularly focusing on improving the fidelity of reconstructed secret images. The original generator and discriminator architectures of CycleGAN were retained to preserve the bidirectional mapping between domains. Additionally, an auxiliary generator-discriminator pair, based on the U-Net architecture ([Ronneberger et al. 2015](#)), was integrated into the framework. This supplementary U-Net-based pathway was introduced to enhance the reconstruction quality of the secret images by leveraging its strong localization and feature preservation capabilities.

These architectural modifications and training enhancements were designed to ensure that the stego images remain visually indistinguishable from normal low-resolution images, while still allowing accurate recovery of the original high-resolution secrets.

4.1.2.1 Generator and Discriminator Architectures

The generator and discriminator models used in the experiments were based on both CycleGAN and U-Net architectural principles. The standard CycleGAN generator is composed of convolutional layers followed by residual blocks and upsampling layers, facilitating effective domain translation. For further enhancement of image reconstruction quality, a U-Net generator was also introduced, which leverages skip connections to retain fine-grained spatial features across layers.

The discriminators used in all experiments follow the PatchGAN structure, which operates on local image patches rather than the full image, thereby enabling better learning of high-frequency details and textures. This setup proved to be effective in improving perceptual quality and guiding the generator towards more realistic outputs.

CycleGAN Generator Architecture

The CycleGAN generator architecture used in this study is designed to facilitate image-to-image translation by learning a mapping between the source and target image domains. It consists of three main stages: an initial convolutional block, a series of residual blocks, and a final upsampling block. The architecture is fully convolutional and maintains the spatial resolution of the input through the use of padding and skip connections.

The initial block includes a 7×7 convolutional layer followed by instance normalization and ReLU activation, which helps capture low-level features. This is followed by two downsampling layers with 3×3 convolutions and stride 2 to reduce spatial dimensions while increasing the depth of feature maps. The core of the generator consists of several residual blocks, each comprising convolutional layers, normalization, and skip connections, which allow the model to learn more complex transformations while preserving information.

In the final stage, the network performs upsampling using transposed convolutions upsampling followed by convolution, effectively restoring the original resolution. The output is passed through a tanh activation function to produce the final image.

Figure 4.11 illustrates the detailed layer-wise architecture of the CycleGAN generator used in our experiments.

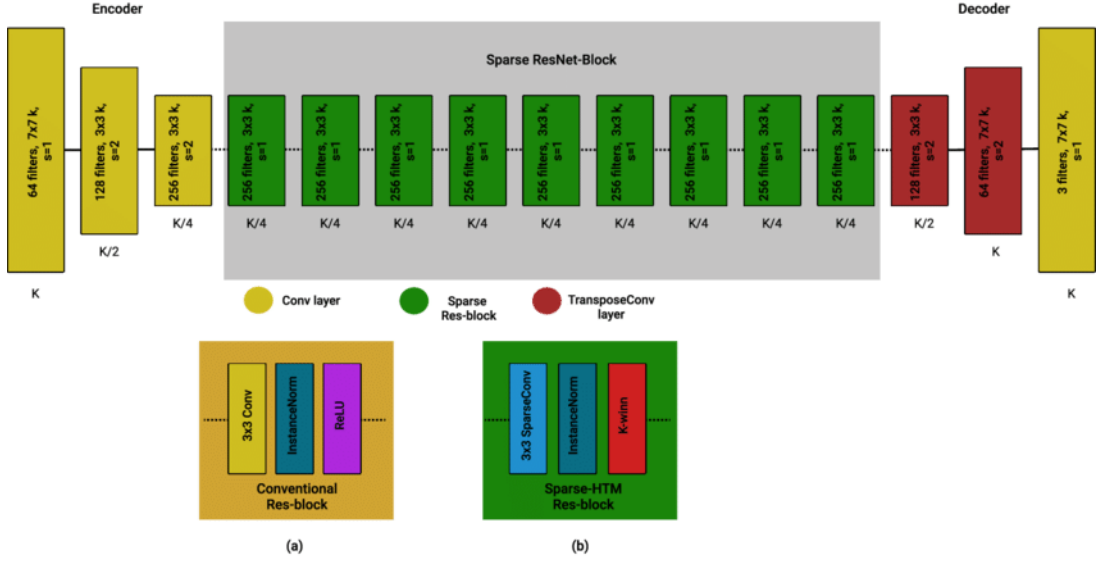


Figure 4.11: CycleGAN Generator Architecture

CycleGAN Discriminator Architecture (PatchGAN)

The discriminator architecture used in CycleGAN follows the PatchGAN design, proposed by [Demir & Unal \(2018\)](#) which classifies image patches rather than the entire image. This approach allows the model to focus on local features, effectively capturing high-frequency details such as textures and edges, which are crucial for ensuring the realism of generated images.

The PatchGAN discriminator operates by sliding a convolutional window over the image and making a real/fake decision for each $N \times N$ patch, rather than producing a single scalar output. This leads to a grid of outputs where each value represents the probability of the corresponding patch being real. This architecture significantly reduces the number of parameters compared to a full-image discriminator, while still being highly effective in guiding the generator towards producing more realistic and locally consistent outputs.

The discriminator comprises a series of strided convolutional layers, each followed by instance normalization (except for the first layer) and a LeakyReLU activation function. These layers progressively reduce the spatial dimensions and increase the depth, ultimately producing an output feature map representing patch-wise classifications.

Figure [4.12](#) illustrates the structure of the PatchGAN discriminator employed in our implementation.

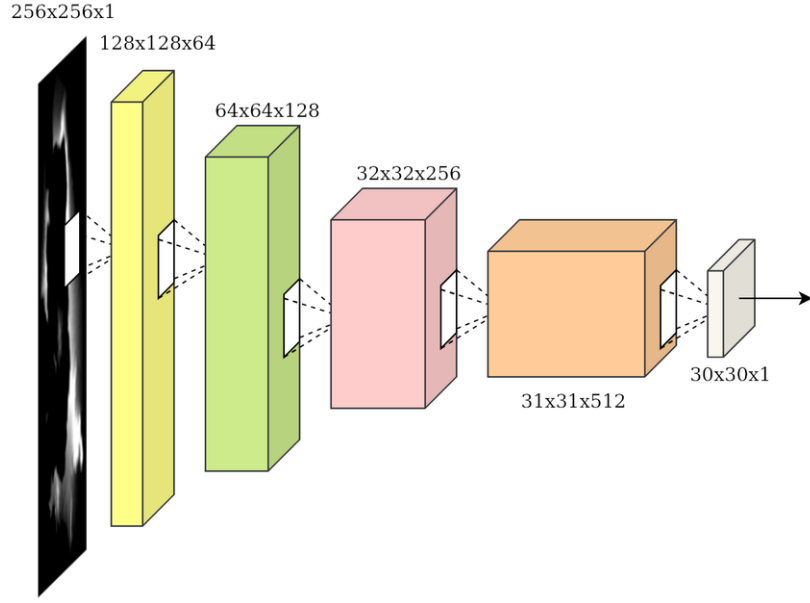


Figure 4.12: PatchGAN Discriminator Architecture

U-Net Generator Architecture

The U-Net generator architecture proposed by [Ronneberger et al. \(2015\)](#) is designed to enhance the CycleGAN framework by introducing a more robust mechanism for preserving spatial details during image transformation. The U-Net architecture is characterized by its encoder-decoder structure with skip connections, which enable it to retain low-level spatial features from the input image while also allowing it to capture high-level abstractions in the deeper layers.

1. **Encoder (Downsampling Path):** This part of the network reduces the spatial dimensions of the input image using convolutional layers followed by instance normalization and activation functions (typically LeakyReLU). Each downsampling step progressively extracts features while halving the spatial dimensions.
2. **Decoder (Upsampling Path):** The decoder upsamples the feature maps to the original image size using transposed convolutions, followed by convolution layers. Skip connections from the encoder path are concatenated with the corresponding upsampled feature maps to retain important low-level features that are otherwise lost in typical downsampling layers. This helps in recovering fine-grained image details.

The U-Net generator benefits from these skip connections by allowing for the

reconstruction of high-resolution features, resulting in better image quality in the final stego image. By preserving both global structure and local detail, the U-Net model helps improve the accuracy of the secret image recovery from the stego image.

Figure 4.13 illustrates the detailed architecture of the U-Net generator used in this study.

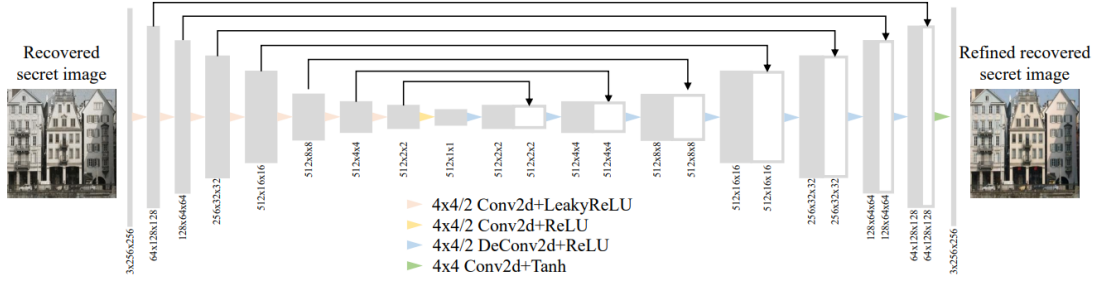


Figure 4.13: U-Net Generator Architecture

4.1.2.2 Customized Loss Function for Stego Image Generation and Enhancement

To effectively train the stego image generation pipeline introduced in this study, a novel loss formulation was designed by combining multiple objectives into a weighted sum. The proposed loss function encourages not only pixel-wise similarity between the reconstructed and original images, but also perceptual quality and adversarial realism. This combination helps achieve high-fidelity reconstructions from visually unrelated stego images.

The generalized form of loss function used is given by:

$$\mathcal{L}_G = \lambda_{\text{rec}} \cdot \mathcal{L}_{\text{rec}} + \lambda_{\text{perc}} \cdot \mathcal{L}_{\text{percep}} + \lambda_{\text{adv}} \cdot \mathcal{L}_{\text{adv}}$$

Where:

- \mathcal{L}_{rec} is the pixel-wise reconstruction loss (L1 loss).
- $\mathcal{L}_{\text{percep}}$ is the perceptual loss computed using feature/style representations.
- \mathcal{L}_{adv} is the adversarial loss.
- $\lambda_{\text{rec}}, \lambda_{\text{perc}}, \lambda_{\text{adv}}$ are scalar weights controlling the influence of each component.

Based on empirical experimentation without extensive hyperparameter tuning, the values for the loss weights were selected.

The training procedure is divided into two distinct stages, each with a tailored loss configuration:

- **Stage 1 (Stego Generation and Reconstruction):** In this stage, $\text{Generator}_{\text{AB}}$, $\text{Generator}_{\text{BA}}$, and $\text{Discriminator}_{\text{ST}}$ are trained jointly. The objective is to generate a realistic stego image and successfully reconstruct the secret image at the receiver end. The generator loss function is formulated as:

$$\mathcal{L}_{\text{Stage1.G}} = 10 \cdot \mathcal{L}_{\text{Recon}} + 5 \cdot \mathcal{L}_{\text{Perceptual}} + \mathcal{L}_{\text{Adv}}$$

- **Stage 2 (Enhancement of Reconstructed Image):** $\text{Generator}_{\text{EN}}$ and $\text{Discriminator}_{\text{EN}}$ are trained to enhance the visual quality of the reconstructed image. The loss function for this stage gives more weight to perceptual similarity and structural accuracy:

$$\mathcal{L}_{\text{Stage2.G}} = 100 \cdot \mathcal{L}_{\text{Recon}} + 50 \cdot \mathcal{L}_{\text{Perceptual}} + \mathcal{L}_{\text{Adv}}$$

The following subsections detail the individual components of the composite loss function.

Reconstruction Loss ($\mathcal{L}_{\text{Recon}}$)

The reconstruction loss is implemented as the pixel-wise L1 loss between the ground truth secret image and its reconstructed version. It ensures that the structural integrity of the original image is preserved during the forward and backward translations. Given a target image I and a prediction \hat{I} , the reconstruction loss is defined as:

$$\mathcal{L}_{\text{Recon}} = \|I - \hat{I}\|_1$$

This term helps reduce blurriness and enforces pixel-level alignment between the reconstructed image and the original secret.

Perceptual Loss ($\mathcal{L}_{\text{Perceptual}}$)

Perceptual loss is computed using feature maps extracted from a pre-trained convolutional neural network (VGG16). It compares the high-level semantic features of the reconstructed and target images, rather than just pixel-wise differences. This allows the model to maintain texture, edges, and finer details that contribute to human-perceived quality. The loss is formally expressed as:

Let:

- I be the generated (input) image.
- T be the target (ground truth) image.
- $\phi_i(\cdot)$ be the activation output from the i -th block of a pre-trained VGG16 network.
- $\mathcal{L}_1(a, b) = \|a - b\|_1$ denote the L1 loss between feature maps.
- $G(\cdot)$ denote the Gram matrix computation.

The total perceptual loss is given by:

$$\mathcal{L}_{\text{Perceptual}}(I, T) = \sum_{i \in \mathcal{F}} \mathcal{L}_1(\phi_i(I), \phi_i(T)) + \sum_{j \in \mathcal{S}} \mathcal{L}_1(G(\phi_j(I)), G(\phi_j(T)))$$

Where:

- \mathcal{F} is the set of feature layers used for content similarity.
- \mathcal{S} is the set of style layers used for texture/style similarity.

The Gram matrix $G(\phi_j(I))$ captures the style representation of an image. For a feature map $\phi_j(I) \in \mathbb{R}^{C \times H \times W}$, it is first reshaped into $\mathbb{R}^{C \times (HW)}$, then:

$$G(\phi_j(I)) = \phi_j(I) \cdot \phi_j(I)^\top \in \mathbb{R}^{C \times C}$$

Adversarial Loss (\mathcal{L}_{Adv})

The adversarial loss is derived from a standard GAN objective, where the generator tries to produce images indistinguishable from real ones, and the discriminator attempts to classify real versus fake samples.

The objective is defined as a minimax game between the generator G and the discriminator D :

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Where:

- G represents the generator network (e.g., $G_{\text{AB}}, G_{\text{BA}}, G_{\text{EN}}$ in our model).
- D denotes the discriminator network (e.g., $D_{\text{ST}}, D_{\text{EN}}$ in our model).
- $x \sim p_{\text{data}}(x)$ is a sample drawn from the real data distribution (e.g., real stego or enhanced images).
- $z \sim p_z(z)$ is the latent input (e.g., secret or stego image) to the generator.
- $D(x)$ is the discriminator's estimated probability that x is real.
- $D(G(z))$ is the probability that the discriminator classifies the generated sample as real.

This loss encourages the generator to produce images that are visually indistinguishable from real samples, effectively learning the data distribution and increasing the quality and realism of the stego and reconstructed images in our model.

4.1.2.3 Experiment 1: Stego Image Generation with Same Resolution as Secret Image

In this experiment, we present a stego image generation pipeline that aims to embed a high-resolution secret image within a visually similar carrier-like image, referred to as the stego image, while maintaining the same resolution as the original secret. The objective is to ensure that the generated stego image appears visually unrelated to the original secret image, while still allowing for effective reconstruction and enhancement at the receiver side.

The proposed architecture is composed of three generator networks and two discriminator networks. A high-level overview of the system is shown in Figure 4.14. Each component of the pipeline plays a specific role in the encoding, reconstruction, and enhancement processes.

1. **Generator_{AB} (Stego Generator):** This generator takes the original secret image I_S as input and produces a visually unrelated stego image I_{ST} . The aim is to hide the high-frequency content of the secret image while preserving latent features necessary for future reconstruction. It functions as the forward translation module from the secret domain to the stego domain.
2. **Discriminator_{ST} (Stego Discriminator):** This discriminator evaluates whether an image in the stego domain is real or generated. It guides Generator_{AB} to produce realistic-looking stego images that are indistinguishable from real images in the stego domain.
3. **Generator_{BA} (Reconstruction Generator):** This generator acts at the receiver side, attempting to reconstruct the secret image \hat{I}_S from the generated stego image I_{ST} . It serves as the reverse translation module that aims to recover the embedded information as accurately as possible.
4. **Generator_{EN} (Enhancement Generator):** To further improve the quality and visual fidelity of the reconstructed image, a U-Net based enhancement generator is employed. This network takes \hat{I}_S as input and outputs an enhanced image \hat{I}_S^+ that is closer in quality and detail to the original secret.
5. **Discriminator_{EN} (Enhancement Discriminator):** This discriminator is responsible for assessing the realism of the enhanced reconstructed image. It ensures that Generator_{EN} produces high-quality outputs that are indistinguishable from real high-resolution secret images.

The system is trained using a combination of adversarial losses, cycle-consistency losses, and reconstruction losses to ensure the fidelity of both the stego image and the reconstructed secret. The enhancement generator introduces a refinement step, which particularly benefits edge recovery and fine detail preservation.

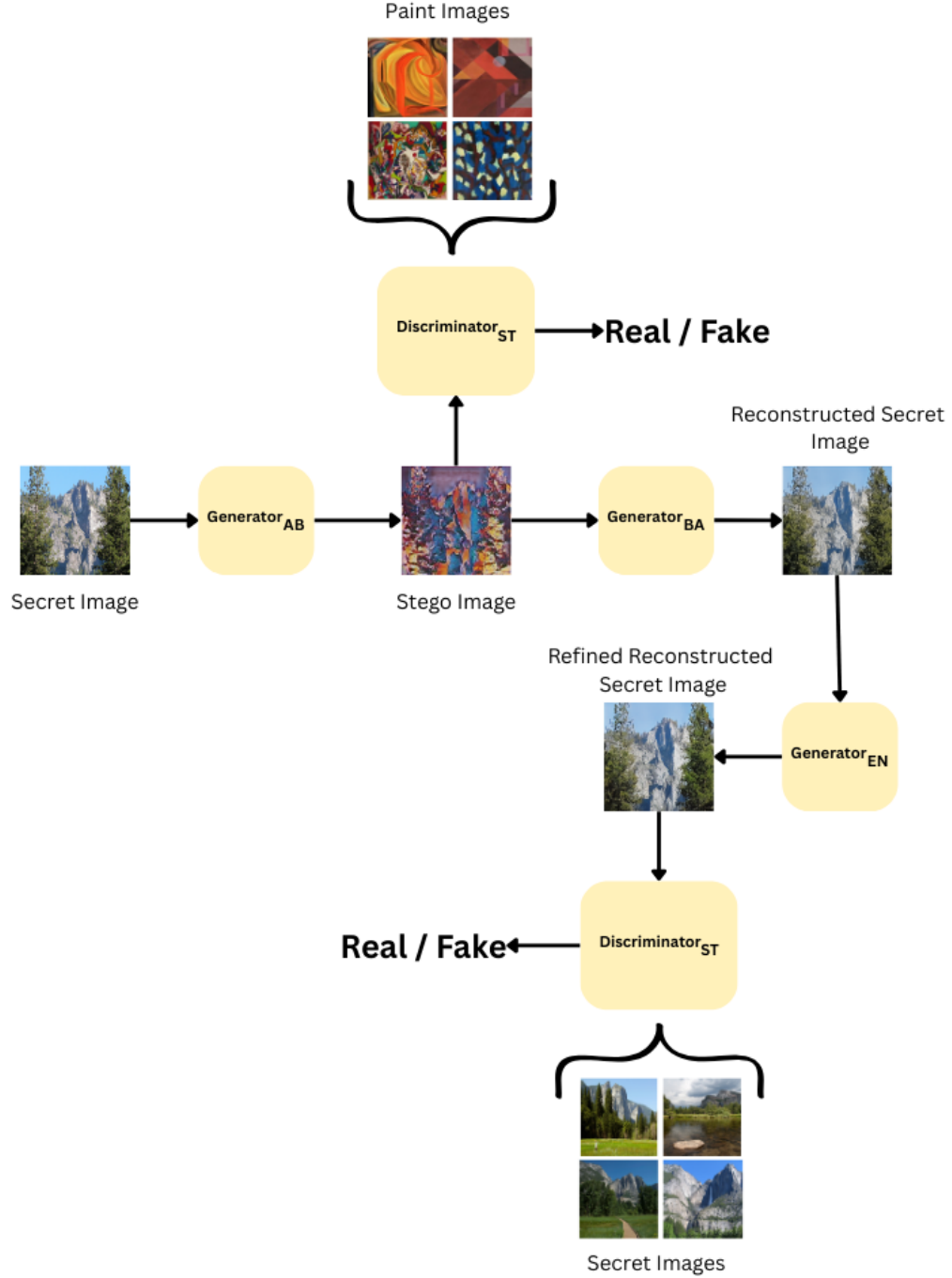


Figure 4.14: High-Level Architecture of Stego Image Generation and Reconstruction Pipeline for Experiment 1.

To evaluate the effectiveness of the proposed stego image generation and reconstruction pipeline, a qualitative analysis was conducted using visual outputs from the trained model. Figure 4.15 illustrates the transformation process across multiple samples.

- **Secret Image:** The secret image that is intended to be hidden.
- **Stego Image:** The visually unrelated image produced by GeneratorAB. Although perceptually different from the secret, it carries the latent features required for reconstruction.
- **Reconstructed Secret Image:** The output of GeneratorBA, which attempts to retrieve the original secret content from the stego image.
- **Refined Reconstructed Secret Image:** The final output after passing the reconstructed image through the enhancement network (GeneratorEN), aiming to recover fine details and visual quality.

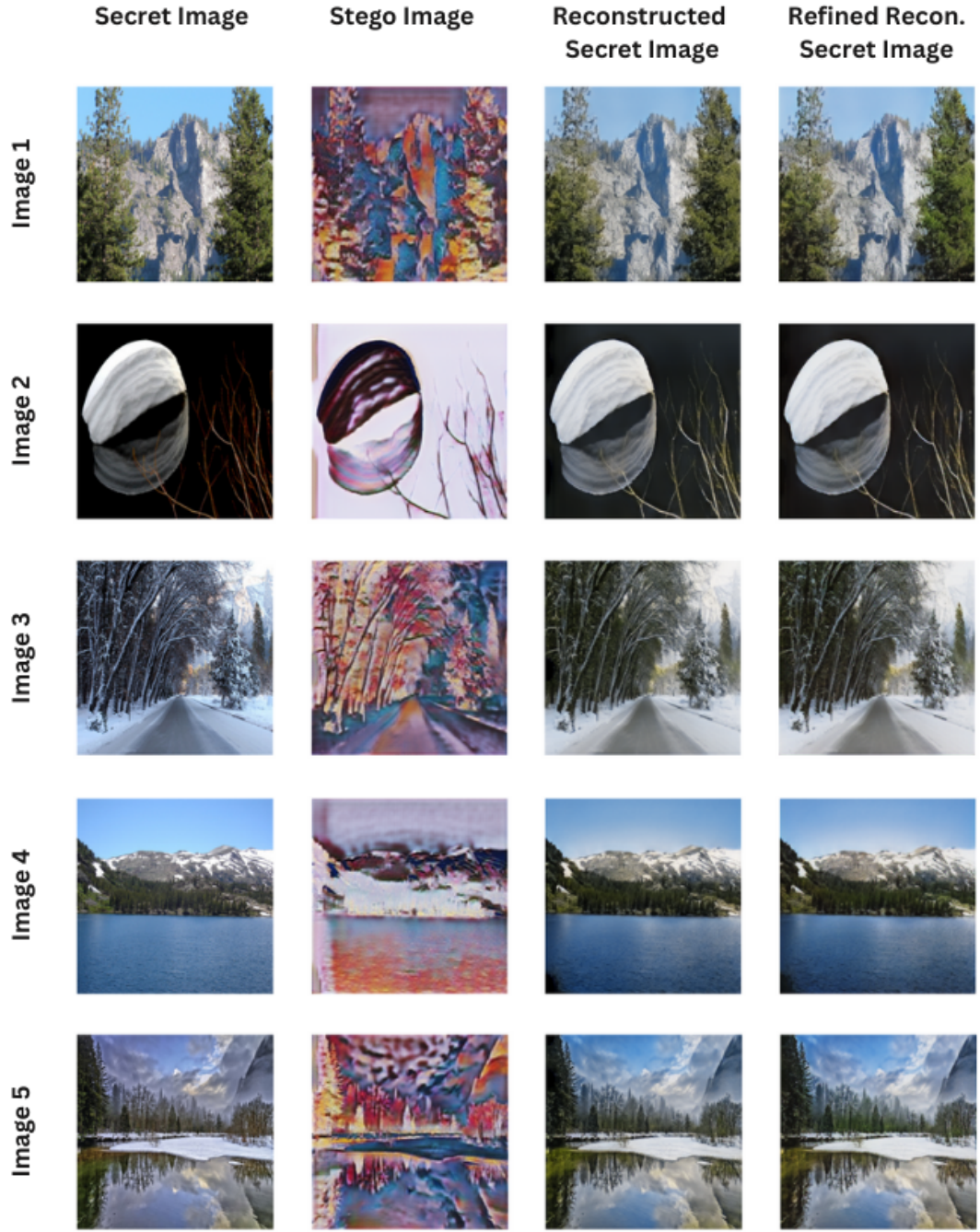


Figure 4.15: Qualitative results for Experiment 1 showing the Secret Image, Stego Image, Reconstructed Secret Image, and Refined Reconstructed Secret Image for several test cases.

The results demonstrate that the proposed architecture is capable of generating realistic stego images with the same resolution as the secret image, that are

visually distinct from the secret images while still enabling successful reconstruction and enhancement. The enhancement step notably improves perceptual clarity and sharpness in the reconstructed images.

Discussion

The architecture presented in this experiment serves as the baseline model for all subsequent stego image generation experiments. In the following sections, we explore how this architecture can be extended to generate low-resolution stego images by integrating various image downsampling and restoration techniques discussed previously in Section [4.1.1](#).

These enhancements aim to achieve a novel objective: embedding high-resolution secret images into low-resolution stego images, while preserving the ability to reconstruct high-quality approximations of the original secret. By combining the foundational framework described above with the transformation and restoration strategies explored earlier, we propose a new class of coverless image steganography models capable of achieving secure, efficient, and resolution-aware information hiding.

4.1.2.4 Experiment 2: Low-Resolution Stego Image Generation Using Bicubic Interpolation

In this experiment, we extend the baseline stego image generation pipeline introduced in Experiment 1 by incorporating a low-resolution generation strategy. The objective is to explore the feasibility of embedding high-resolution secret images into visually unrelated low-resolution stego images, which can be effectively transmitted or stored with reduced bandwidth and then reconstructed at the receiver side.

Motivation. This design decision is motivated by the high performance of the bicubic interpolation technique, as observed in the high-resolution restoration experiments discussed in Section 4.1.1. Among various methods evaluated, bicubic interpolation yielded the most satisfactory balance between computational efficiency and reconstruction quality, making it a suitable candidate for use in this context.

Architectural Overview. The modified architecture for this experiment is illustrated in Figure 4.16 and consists of the following stages:

1. **Low-Resolution Preprocessing:** The original high-resolution secret image $I_S^{HR} \in \mathbb{R}^{512 \times 512}$ is first downsampled using bicubic interpolation to obtain a low-resolution representation $I_S^{LR} \in \mathbb{R}^{256 \times 256}$. This step significantly reduces the image size while preserving structural integrity.
2. **Stego Image Generation (GenAB):** The low-resolution secret image I_S^{LR} is then fed into the GeneratorAB network, which generates a low-resolution stego image I_{ST}^{LR} . This image is designed to be visually unrelated to the secret while encoding essential latent features necessary for reconstruction.
3. **Reconstruction (GenBA):** At the receiver side, the GeneratorBA attempts to recover a low-resolution approximation \hat{I}_S^{LR} of the original secret image from the stego image.
4. **High-Resolution Upsampling:** The reconstructed low-resolution image \hat{I}_S^{LR} is then upsampled back to the original resolution $\hat{I}_S^{HR} \in \mathbb{R}^{512 \times 512}$ using bicubic interpolation, ensuring that the output is suitable for further enhancement and visualization.
5. **Refinement (GenEN):** Finally, the upsampled image \hat{I}_S^{HR} is passed through the U-Net-based enhancement generator (GenEN), producing the refined high-resolution secret image \hat{I}_S^{+HR} , which more closely resembles the original secret image in terms of both texture and perceptual fidelity.

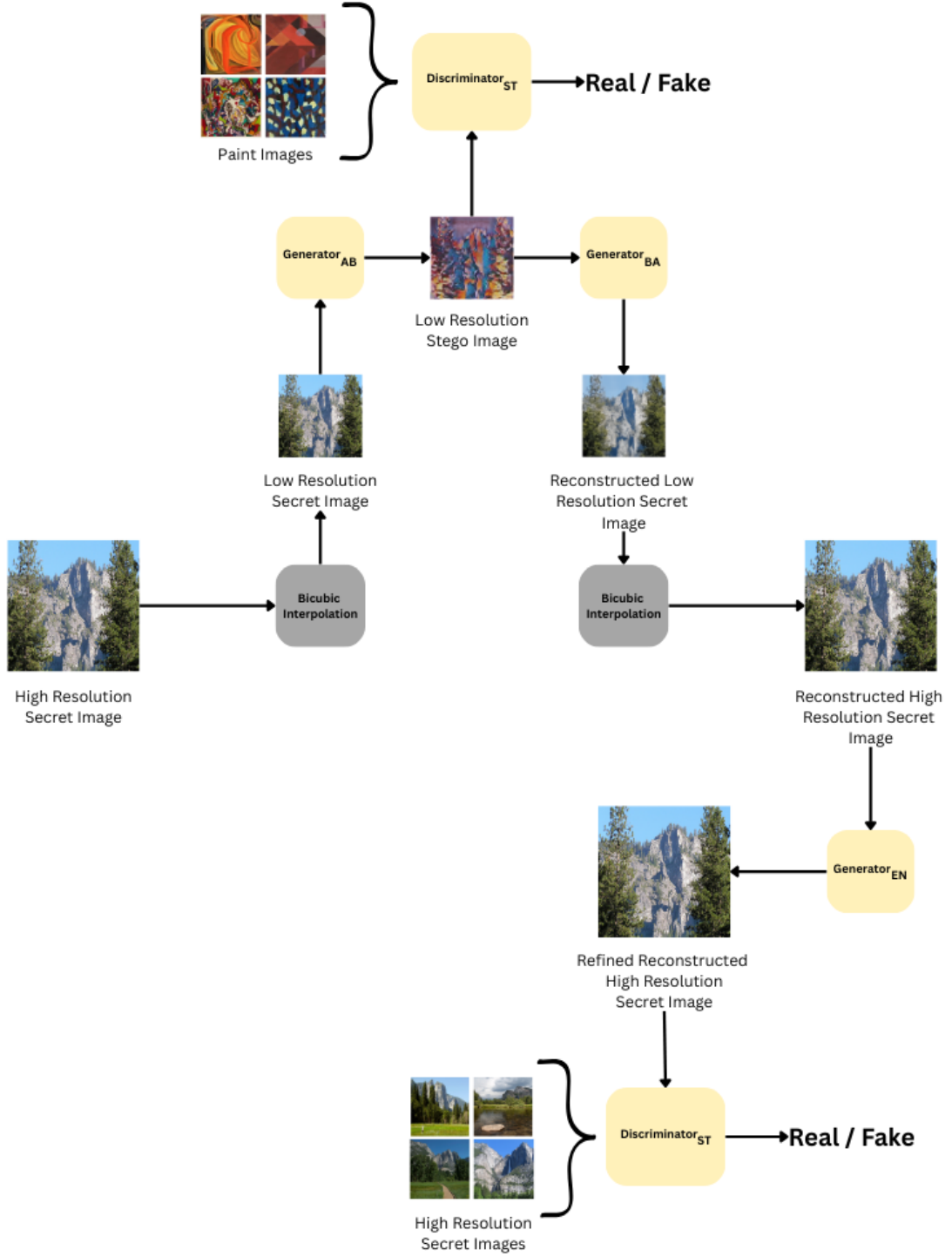


Figure 4.16: High-Level Architecture for Experiment 2: Low-Resolution Stego Image Generation Pipeline Incorporating Bicubic Interpolation.

Figure 4.17 illustrates the results obtained from the proposed low-resolution

stego image generation and reconstruction pipeline. The visual outputs demonstrate the effectiveness of the method in concealing high-resolution information within a downsampled stego image and subsequently reconstructing it with notable fidelity.

The columns in the figure represent the following stages:

1. **HR Secret Image:** The original high-resolution secret image used as the input.
2. **LR Secret Image:** The downsampled version of the secret image using bicubic interpolation.
3. **LR Stego Image:** The visually unrelated stego image generated from the LR secret image by GeneratorAB.
4. **Reconstructed LR Secret Image:** The recovered LR image produced by GeneratorBA.
5. **Reconstructed HR Secret Image:** The upsampled reconstruction using bicubic interpolation.
6. **Refined HR Secret Image:** The final enhanced output obtained from the U-Net based GeneratorEN.

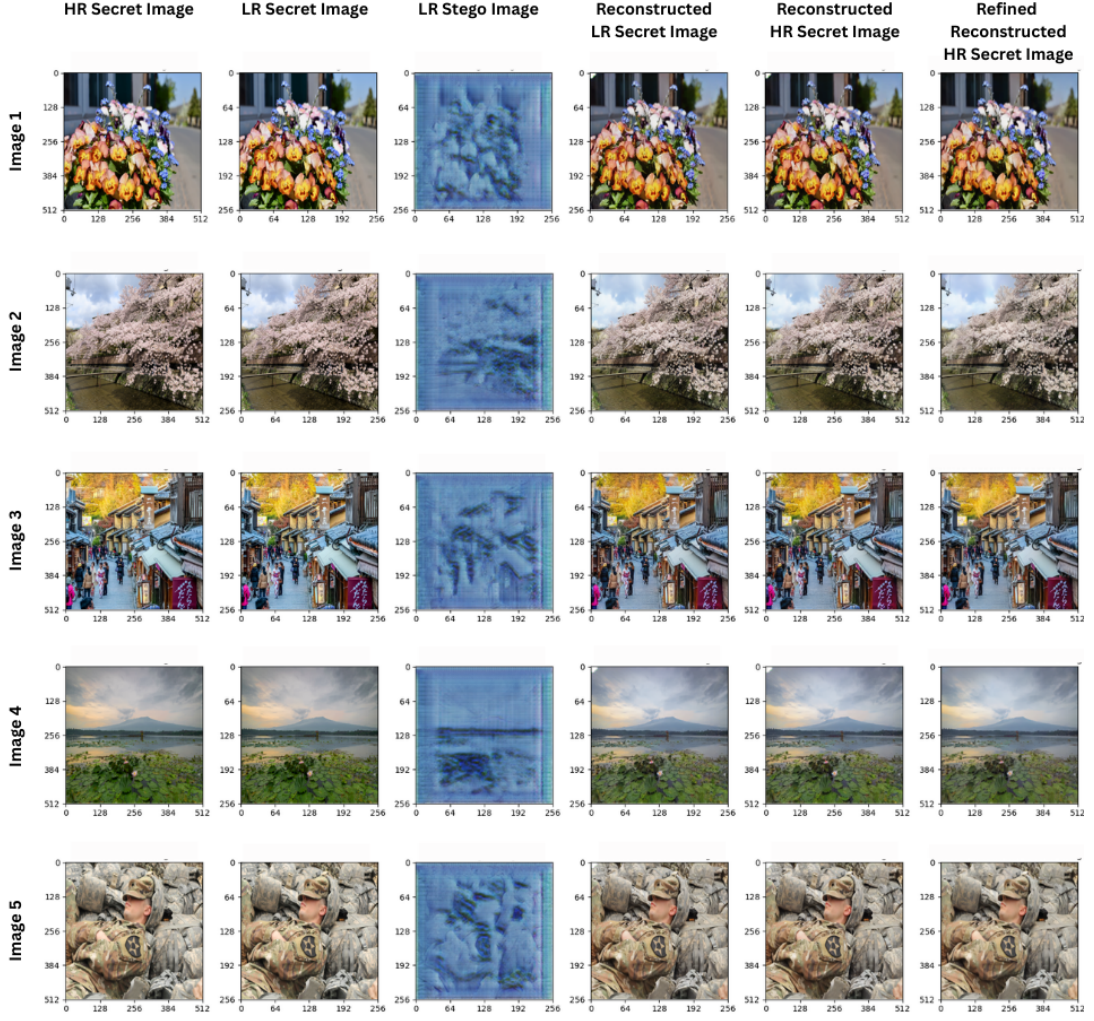


Figure 4.17: Visual results from Experiment 2. Each row corresponds to one test sample, progressing from the original HR secret image to the final refined reconstruction.

As seen in the results, the reconstructed images retain structural and perceptual similarity to the original high-resolution secret, despite the transformations and resolution changes applied during the steganographic process. The enhancement stage notably improves sharpness and detail, validating the contribution of GeneratorEN.

Discussion

While the proposed low-resolution stego generation pipeline performs effectively in preserving and reconstructing complex secret images, several limitations were observed with simpler secret images. Specifically, when the secret image contains

only a single or a few prominent objects with relatively plain backgrounds, the generated stego images tend to visually retain structural cues of the original content. This compromises the visual dissimilarity requirement expected in a steganographic system.

Figure 4.18 illustrates this phenomenon. As seen in the sample results, the generated stego images exhibit noticeable traces of the original secret object, making them potentially vulnerable to detection or interpretation.

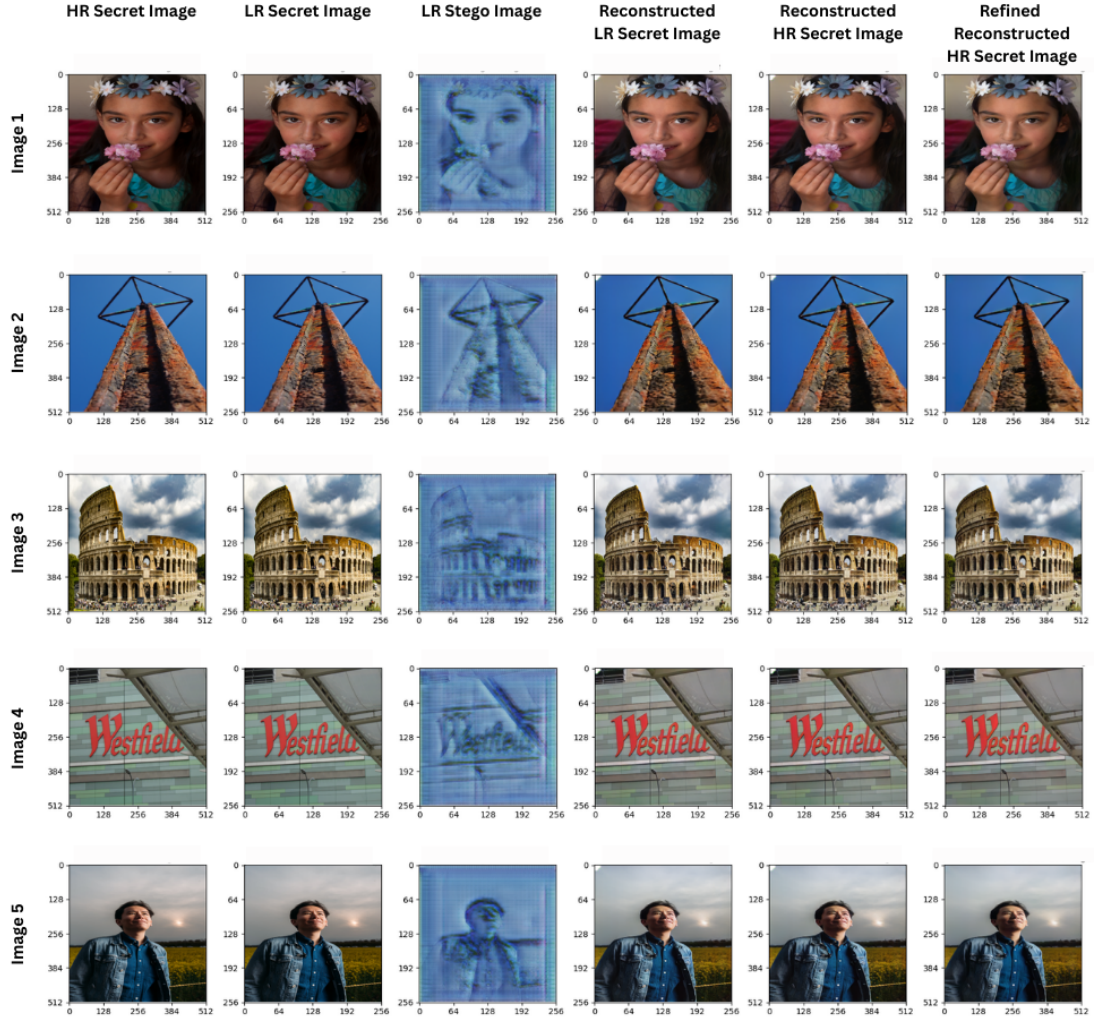


Figure 4.18: Examples demonstrating limitations of Experiment 2 for simple secret images. The stego images in these cases still reveal perceptual hints of the secret content.

This observation indicates that the architecture in Experiment 2 is better suited for more complex secret images with cluttered scenes, where the transformation

to the stego domain effectively hides individual object features among dense background information.

To address this challenge, the following experiments were designed to introduce architectural and methodological enhancements aimed at improving the concealment capabilities for simpler, object-focused secret images.

4.1.2.5 Experiment 3: Auto-Encoder-Based Low-Resolution Stego Image Generation

Following the limitations observed in Experiment 2, this experiment was designed to improve concealment capabilities, especially for simpler, object-focused secret images. The key issue identified was that when the secret image contains only a few salient objects, the generated stego image may still preserve recognizable visual cues, thereby weakening the steganographic effectiveness.

To address this challenge, we proposed using the **Auto-Encoder Based High-Resolution Image Restoration** method (previously discussed in Section 4.1.1) for generating the low-resolution version of the secret image. The motivation behind this choice is that the auto-encoder tends to produce a "washed-out" version of the secret, where high-frequency details and textures are suppressed. This is illustrated in Figure 4.19, where the low-resolution output exhibits minimal visual resemblance to the original image.

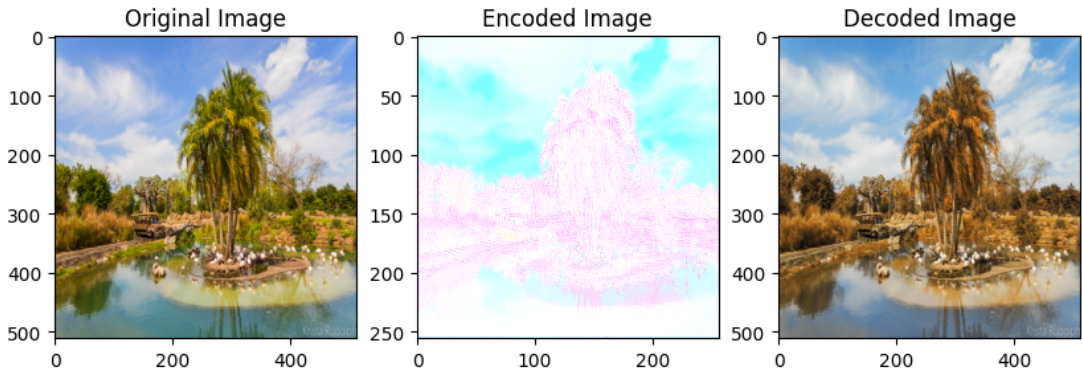


Figure 4.19: Comparison between high-resolution secret image and the low-resolution image generated via auto-encoder. Note the significant loss of visual detail, which enhances concealment.

Hypothesis

The hypothesis for this experiment is that by feeding a visually non-informative, low-resolution image (generated by the auto-encoder) into the stego generation pipeline, the resulting stego image will effectively obscure any identifiable features of the original secret. At the same time, we expect that sufficient latent features are preserved for reconstructing the original image with reasonable accuracy.

High-Level Architecture

The architecture of the system is composed of the following stages:

1. **Encoding (Downsampling):** The original high-resolution secret image (I_S) is passed through a trained encoder to generate a low-resolution representation (I_{LR}) of size 256×256 .
2. **Stego Generation (GeneratorAB):** The low-resolution secret image (I_{LR}) is then used as input to GeneratorAB, which produces a stego image (I_{ST}) in the same low-resolution space.
3. **Reconstruction (GeneratorBA):** GeneratorBA receives the low-resolution stego image and attempts to reconstruct the low-resolution secret image (\hat{I}_{LR}).
4. **Decoding (Upsampling):** The reconstructed low-resolution secret image is then passed through a decoder (paired with the encoder used in step 1) to upscale it back to the high-resolution space (\hat{I}_S).
5. **Enhancement (GeneratorEN):** Finally, the reconstructed high-resolution image is passed through the enhancement generator (UNet-based) to recover fine details and improve visual quality (\hat{I}_S^+).

The complete high-level system is illustrated in Figure [4.20](#).

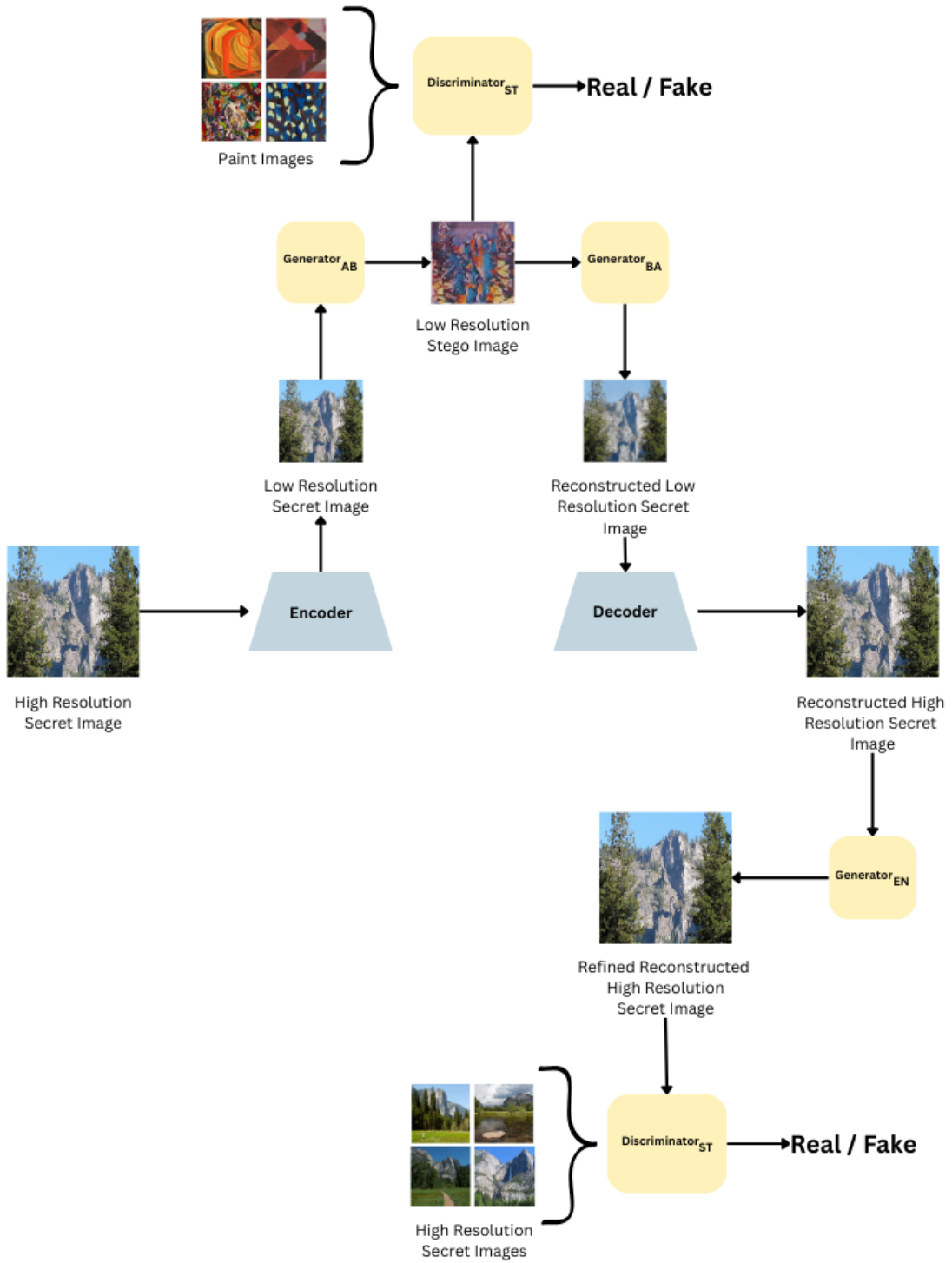


Figure 4.20: High-Level Architecture of Auto-Encoder-Based Low-Resolution Stego Image Generation Pipeline (Experiment 3).

While this model successfully generated stego images that visually obscure nearly

all meaningful features of the original secret—achieving the intended goal of enhanced concealment—it suffered from severe reconstruction degradation. The decoder and enhancement generator were not able to recover the fine-grained image details from the auto-encoded low-resolution input, leading to blurred and perceptually degraded outputs.

Figure 4.21 illustrates the outcome of this experiment, demonstrating that although the stego image appears effectively unrelated to the original secret, the reconstructed and refined versions fail to resemble the source image in meaningful ways.

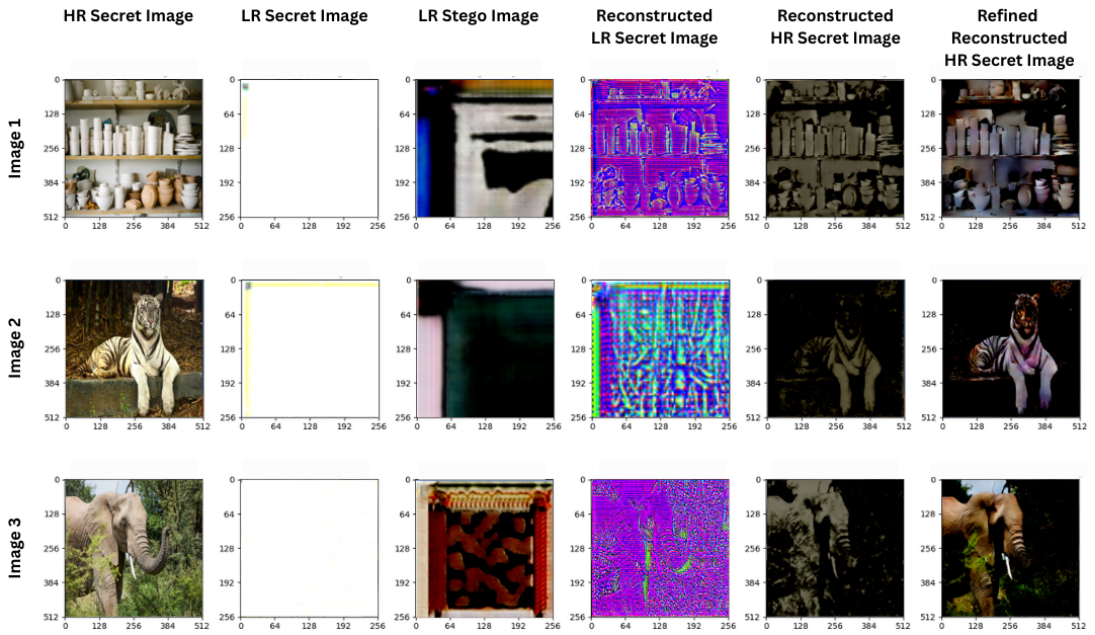


Figure 4.21: Experiment 3 Results. Columns from left to right: High-Resolution Secret Image, Auto-Encoded LR Image, Stego Image, Reconstructed LR Image, Reconstructed HR Image, Refined HR Image.

Discussion

The experiment demonstrates that while auto-encoder-based downsampling significantly improves concealment, it sacrifices reconstructive accuracy. As such, this approach was deemed unsuitable for scenarios where high-fidelity recovery of the original image is essential. Subsequent experiments explore alternative approaches to balance concealment and reconstruction quality more effectively.

4.1.2.6 Experiment 4: Scrambling-Based Concealment Prior to Stego Generation

After the failure of Experiment 3, which prioritized concealment at the expense of reconstruction quality, this experiment explores an alternative strategy aimed at maintaining both visual obfuscation and high reconstruction fidelity. The approach involves scrambling the low-resolution secret image before feeding it to the stego generator, and then applying inverse scrambling after reconstruction. The intuition is that scrambling spatially disrupts the secret image, making it visually incomprehensible, while preserving pixel-level information that can still be learned and decoded by the generator networks.

Unlike the auto-encoder used in Experiment 3—which suppressed too much visual and structural information—scrambling serves as a reversible transformation that hides spatial patterns without degrading the information content. This enables the system to generate stego images that are independent of the original secret’s appearance while still allowing for accurate downstream reconstruction.

Scrambling and Inverse Scrambling Mechanism

To enhance concealment and disrupt the spatial correlation of pixels prior to steganographic embedding, we employed a mathematical scrambling technique. This scrambling is performed in the spatial domain using linear transformations derived from modular matrix arithmetic, inspired by Arnold Cat Map-like transformations.

Scrambling Operation Let the input image tensor be represented as $\mathbf{I} \in \mathbb{R}^{B \times C \times H \times W}$, where $H = W = N$ (i.e., the image must be square). The scrambling is applied independently across each image in the batch.

1. Generate a coordinate grid for pixel locations:

$$\mathbf{X}, \mathbf{Y} = \text{meshgrid}(\{0, 1, \dots, N - 1\})$$

2. Stack the coordinates into a 2D matrix:

$$\mathbf{P} = \begin{bmatrix} x_1 & x_2 & \dots & x_{N^2} \\ y_1 & y_2 & \dots & y_{N^2} \end{bmatrix} \in \mathbb{Z}^{2 \times N^2}$$

3. Define a transformation matrix \mathbf{A} :

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

4. Apply the scrambling transformation iteratively:

$$\mathbf{P}' = (\mathbf{A}^k \cdot \mathbf{P}) \mod N$$

where k is the number of scrambling iterations (default is 1), and operations are performed modulo N to ensure valid pixel positions.

5. Rearrange the pixels of the image tensor based on the transformed coordinates \mathbf{P}' to obtain the scrambled image.

Inverse Scrambling Operation To recover the original pixel arrangement, the inverse scrambling applies the inverse of matrix \mathbf{A} , denoted \mathbf{A}^{-1} , also in modular arithmetic:

$$\mathbf{A}^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

Applying this inverse transformation k times undoes the scrambling:

$$\mathbf{P} = (\mathbf{A}^{-k} \cdot \mathbf{P}') \mod N$$

The recovered coordinate grid \mathbf{P} is then used to reorder the scrambled image pixels back to their original positions, effectively reconstructing the initial spatial structure.

Figure [4.22](#) illustrates the visual output of a sample scrambled image. The scrambling process significantly disrupts spatial coherence, making it difficult to discern any recognizable features or patterns from the original image.

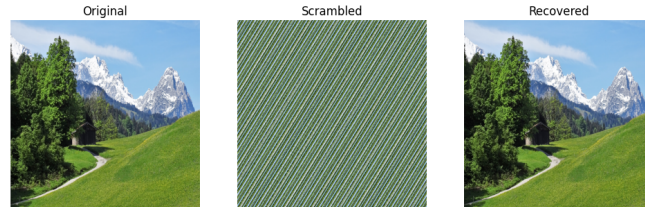


Figure 4.22: Visual output of a sample scrambled image.

Benefits This scrambling approach:

- Prevents trivial visual cues in the stego image by disrupting global spatial continuity.
- Provides an additional layer of security as the transformation is reversible only with knowledge of the matrix and the number of iterations.
- Is computationally lightweight and fully differentiable, enabling integration into end-to-end neural networks.

Thus, scrambling serves as a pre-processing mechanism to increase concealment efficacy without compromising reconstruction capabilities when used with a corresponding inverse transformation.

High-Level Architecture

The proposed architecture builds upon the pipeline used in Experiment 2, with the key modification being the introduction of a scrambling and inverse scrambling step. The flow of the system is as follows:

1. **Downsampling (Bicubic Interpolation):** The original high-resolution secret image (I_S) is first resized to a low-resolution version (I_{LR}) of size 256×256 using bicubic interpolation.
2. **Scrambling:** A scrambling function $f_{scramble}$ is applied to I_{LR} to generate a spatially randomized image $I_{scrambled}$.
3. **Stego Generation (GeneratorAB):** The scrambled low-resolution image is then used as input to GeneratorAB, which produces a visually unrelated stego image I_{ST} .
4. **Reconstruction (GeneratorBA):** The low-resolution stego image is passed through GeneratorBA, which reconstructs the scrambled secret image $\hat{I}_{scrambled}$.
5. **Inverse Scrambling:** The inverse scrambling function $f_{scramble}^{-1}$ is applied to $\hat{I}_{scrambled}$ to recover the reconstructed low-resolution secret image \hat{I}_{LR} .
6. **Upsampling (Bicubic Interpolation):** The reconstructed low-resolution image is upscaled back to the original resolution using bicubic interpolation, producing \hat{I}_S .

7. **Enhancement (GeneratorEN):** Finally, the upscaled image is passed through the enhancement generator to recover fine details and improve visual fidelity, resulting in the refined image \hat{I}_S^+ .

The high-level architecture is illustrated in Figure [4.23](#).

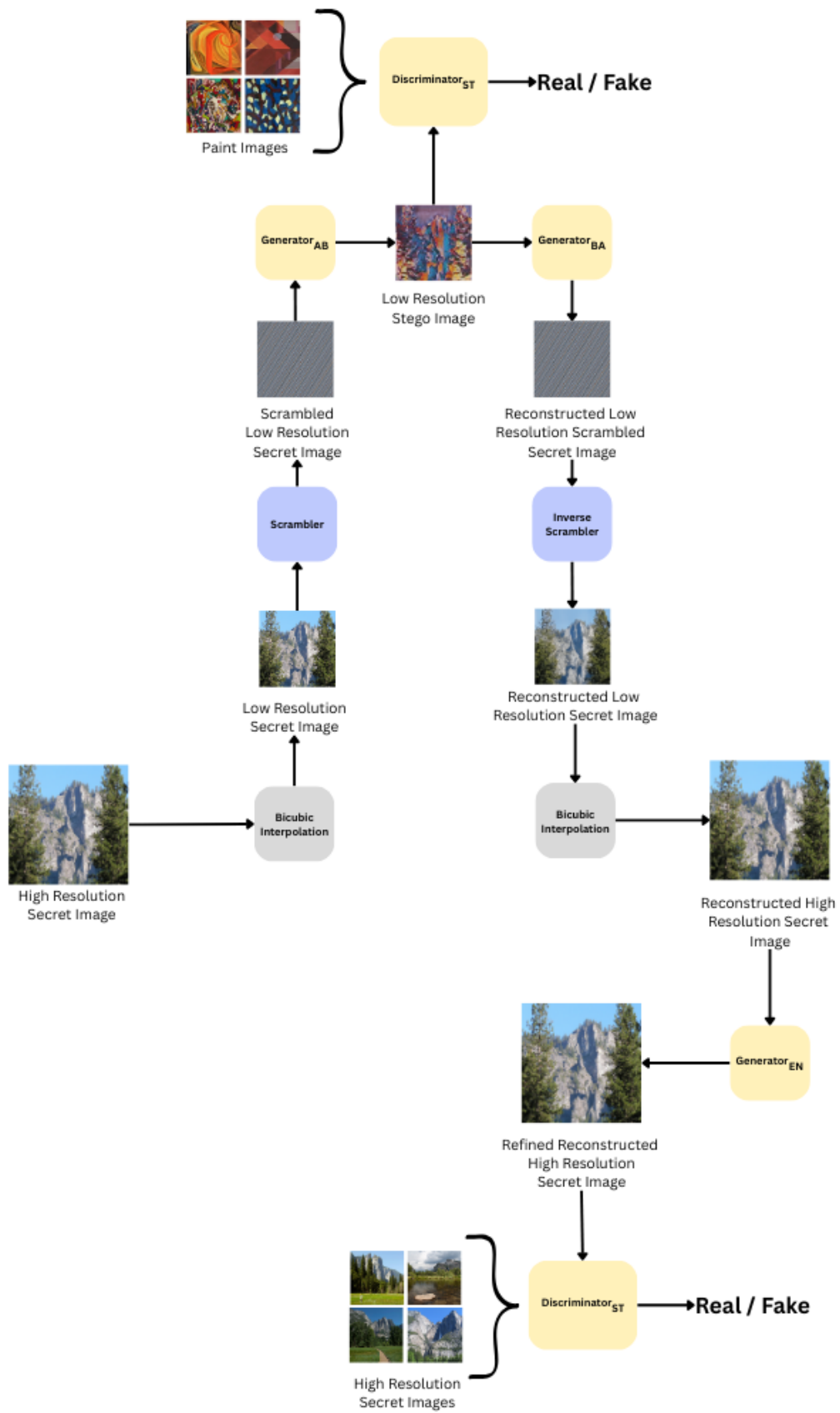


Figure 4.23: High-Level Architecture of the Scrambling-Based Stego Generation and Reconstruction Pipeline (Experiment 4).

This method effectively generated stego images that were visually uncorrelated with the original secret, successfully concealing object-level and texture-based information. As shown in Figure [4.24](#), the scrambling process introduced a sufficient degree of obfuscation, while still enabling GeneratorBA to reconstruct a meaningful representation when paired with inverse scrambling.

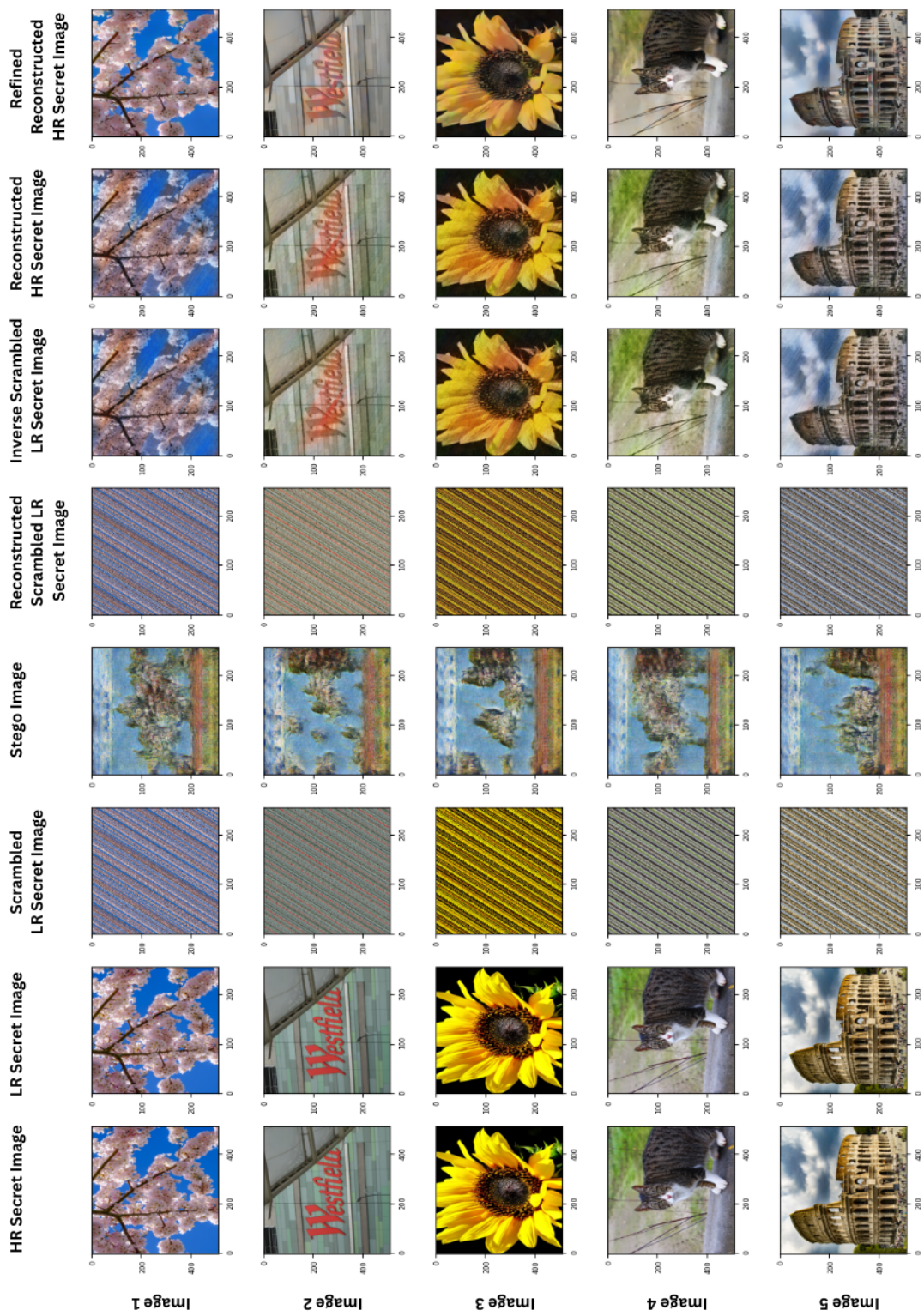


Figure 4.24: Experiment 4 Results. Columns from left to right: HR Secret Image, LR Secret Image, Scrambled LR Image, Stego Image, Reconstructed Scrambled LR Image, Inverse-Scrambled LR Image, Reconstructed HR Image, Refined HR Image.

Discussion

This scrambling-based approach demonstrated a significant advancement in concealment capabilities, which is the primary objective in steganography. By introducing spatial randomness to the secret image before stego generation, the system effectively eliminated visual clues that could reveal the content of the original image. Even for object-focused secret images—where previous experiments failed to hide obvious features—this method successfully produced stego images that were visually independent of the secrets.

Although the reconstruction accuracy did not surpass that of Experiment 2, the concealment quality was notably superior. This trade-off is acceptable and even desirable in many steganographic applications where secrecy takes precedence over perfect reconstruction. Therefore, this experiment marks a promising direction for scenarios that require stronger visual obfuscation without compromising reconstructability beyond usability.

4.1.2.7 Selection of Final Models for Evaluation

The experiments described above explored various strategies for generating stego images from high-resolution secret images while balancing the trade-off between concealment and reconstruction quality. Each experiment introduced different methodologies to either improve the visual independence of the stego image or enhance the quality of the reconstructed secret.

Based on the outcomes, **Experiment 2** and **Experiment 4** were selected as the final models for further evaluation. **Experiment 2** demonstrated superior reconstruction quality, making it ideal for scenarios where recovery of the original content is critical. In contrast, **Experiment 4** achieved stronger concealment, particularly for object-focused secret images, making it more suitable for steganographic applications that prioritize secrecy over reconstruction accuracy.

The next section presents a comprehensive evaluation of these two selected models in terms of:

- **Reconstruction Quality:** How accurately the original secret image can be recovered.
- **Hiding Capacity:** The amount of information that can be embedded without degrading visual quality.
- **Robustness to Steganalysis:** Resistance to detection by modern steganalysis tools.

- **Applicability:** Real-world suitability based on image complexity, performance, and use case requirements.

This evaluation phase provides a deeper insight into the strengths and limitations of each model, guiding their applicability in practical steganographic scenarios.

Chapter 05

5.1 Comprehensive Evaluation of Final Stego Image Generation Models

To thoroughly assess the effectiveness of the proposed stego image generation approaches, this section presents a comprehensive evaluation of the two selected models — **Experiment 2** (Bicubic Interpolation-based Low-Resolution Stego Generation) and **Experiment 4** (Scrambled Input-based Stego Generation). These models were chosen based on their respective strengths in reconstruction fidelity and concealment capabilities.

The evaluation focuses on four key criteria that are critical to steganographic systems:

- **Reconstruction Quality:** Evaluating how accurately the original high-resolution secret image can be reconstructed at the receiver side.
- **Hiding Capacity and Secret Size Reduction:** Measure the amount of information that can be embedded within a cover image and the efficiency gained by reducing the secret image size.
- **Robustness to Steganalysis:** Assessing each model’s resistance to detection by conventional and deep learning-based steganalysis tools.
- **Applicability:** Investigating the practical usability of the models in real-world scenarios based on factors like image complexity, model reliability, and intended use cases.

Through both quantitative metrics and qualitative visual analysis, this evaluation aims to provide clear insight into the trade-offs and strengths of each approach. The ultimate goal is to understand the suitability of each model for different steganographic objectives — whether the focus lies on secrecy, recoverability, or a balance of both.

5.1.1 Qualitative Evaluation: Visual Analysis of Reconstructed Images

5.1.1.1 Quantitative Evaluation: SSIM, PSNR, and MAE

To evaluate the reconstruction quality of the two selected models, we used three widely accepted image quality assessment metrics:

- **SSIM (Structural Similarity Index Measure):** Evaluates perceptual similarity between the original and reconstructed images based on luminance, contrast, and structure.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (5.2)$$

where μ_x , μ_y are the means, σ_x^2 , σ_y^2 are the variances, σ_{xy} is the covariance, and C_1 , C_2 are constants to stabilize the division.

- **PSNR (Peak Signal-to-Noise Ratio):** Measures the ratio between the maximum possible power of an image and the power of noise affecting the quality of its representation.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (5.3)$$

where MAX_I is the maximum possible pixel value (e.g., 255 for 8-bit images), and MSE is the Mean Squared Error between the original and reconstructed images.

- **MAE (Mean Absolute Error):** Computes the average magnitude of pixel-wise differences between the original and reconstructed images.

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \quad (5.4)$$

where x_i and y_i are pixel values from the original and reconstructed images respectively, and N is the total number of pixels.

Table [5.1.1.1](#) summarizes the average metric values obtained from a test set of images for **Experiment 2** and **Experiment 4**.

Experiment	SSIM \uparrow	PSNR (dB) \uparrow	MAE \downarrow
Experiment 1	0.8121	72.63	0.0559
Experiment 3	0.7123	70.89	0.0800

Table 5.8: Quantitative Comparison of Reconstruction Quality

From the results, it is evident that **Experiment 2** provides superior reconstruction quality compared to **Experiment 4**, achieving higher SSIM and PSNR values along with a lower MAE. This aligns with our previous qualitative observations and validates its effectiveness in preserving fine details during the reconstruction process.

To contextualize the effectiveness of the proposed coverless image steganography method, we compared our results with notable prior approaches including Baluja [\(2017\)](#), Li et al. [\(2021\)](#), and Liu et al. [\(2020\)](#). The comparison focuses on widely accepted quantitative metrics such as Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Mean Absolute Error (MAE), as shown in Table [5.1.1.1](#).

Method / Experiment	SSIM \uparrow	PSNR (dB) \uparrow	MAE \downarrow
Baluja (2017)	0.89	27.51	-
Li et al. (2021)	0.73	29.23	-
Liu et al. (2020)	0.97	36.23	-
Experiment 1 (Ours)	0.8121	72.63	0.0559
Experiment 3 (Ours)	0.7123	70.89	0.0800

Table 5.9: Quantitative Comparison of Reconstruction Quality with Prior Work

Experiment 2 achieves a PSNR of **72.63 dB**, significantly surpassing the PSNR values of all the prior works, suggesting excellent reconstruction fidelity. Moreover, it maintains a high SSIM of **0.8121** and a relatively low MAE of **0.0559**, demonstrating a balanced trade-off between quality and stealth.

Experiment 4, while showing slightly lower reconstruction quality with a PSNR of **70.89 dB** and SSIM of **0.7123**, excels in concealment effectiveness due to the additional scrambling mechanism, making it more robust against potential visual or statistical analysis.

5.1.1.2 RGB Histogram Comparison Between Original and Reconstructed Secret Images

To evaluate the reconstruction quality of the proposed models, we conducted an RGB histogram-based analysis between the original high-resolution secret images

and their corresponding reconstructed high-resolution secret images. This analysis highlights how well the color distribution has been preserved across each RGB channel.

We computed the following three metrics for each channel (Red, Green, and Blue):

- **Correlation (\uparrow):** Measures how closely the histograms of the original and reconstructed images align. Higher values indicate greater similarity.

$$\text{Correlation}(H_1, H_2) = \frac{\sum_{i=1}^n (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{\sum_{i=1}^n (H_1(i) - \bar{H}_1)^2} \cdot \sqrt{\sum_{i=1}^n (H_2(i) - \bar{H}_2)^2}} \quad (5.5)$$

where H_1 and H_2 are histogram vectors, and \bar{H}_1, \bar{H}_2 are their respective means.

- **Chi-square Distance (\downarrow):** Quantifies the statistical divergence between the histograms. Lower values signify better reconstruction accuracy.

$$\chi^2(H_1, H_2) = \sum_{i=1}^n \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i) + \epsilon} \quad (5.6)$$

where H_1 and H_2 are the histogram bins and ϵ is a small constant added to avoid division by zero.

- **Earth Mover's Distance (EMD) (\downarrow):** Represents the minimal effort required to transform one histogram into another. Lower EMD values indicate better alignment.

$$\text{EMD}(H_1, H_2) = \frac{\sum_{i=1}^n |CDF_{H_1}(i) - CDF_{H_2}(i)|}{n} \quad (5.7)$$

where CDF_H denotes the cumulative distribution function of histogram H .

Table [5.10](#) presents the average RGB histogram comparison results for Experiment 2 and Experiment 4.

Table 5.10: RGB Histogram Comparison Between Original and Reconstructed Secret Images

Metric	Channel	Experiment 1	Experiment 3	Ideal Value
Correlation \uparrow	Red	0.9910	0.9301	1.0000
	Green	0.9828	0.9395	1.0000
	Blue	0.9474	0.9125	1.0000
Chi-square \downarrow	Red	0.3135	5.8246	0.0000
	Green	0.4212	3.3543	0.0000
	Blue	0.8657	3.6065	0.0000
EMD \downarrow	Red	0.0016	0.0089	0.0000
	Green	0.0013	0.0083	0.0000
	Blue	0.0026	0.0064	0.0000

From the table, we observe that **Experiment 2** achieves higher correlation and lower Chi-square and EMD values across most channels, indicating superior reconstruction fidelity in terms of color distribution. On the other hand, **Experiment 4** shows slightly more histogram deviation, highlighting its prioritization of concealment over exact reconstruction quality.

5.1.1.3 Lab Histogram Comparison Between Original and Reconstructed High-Resolution Secret Images

To assess the reconstruction fidelity of the proposed models, we evaluated the similarity between the original high-resolution secret images and the corresponding reconstructed high-resolution secret images in the perceptually aligned CIE Lab color space. This analysis is critical to understanding how well the models preserve perceptual characteristics after the full steganographic encoding and decoding pipeline.

We used the following histogram-based metrics for evaluation:

- **Correlation** (\uparrow): Indicates how well the distributions of the original and reconstructed images match. Values closer to 1.0 signify strong similarity.
- **Chi-square** (\downarrow): Measures the statistical divergence between the two histograms. Lower values denote better similarity.
- **Earth Mover’s Distance (EMD)** (\downarrow): Represents the minimal effort required to transform one distribution into another. Smaller values reflect closer matches.

Table 5.11: Lab Histogram Comparison Between Original and Reconstructed High-Resolution Secret Images

Metric	Channel	Experiment 1	Experiment 3	Ideal Value
Correlation \uparrow	L (Lightness)	0.9945	0.9627	1.0000
	a (Green–Red)	0.9501	0.9615	1.0000
	b (Blue–Yellow)	0.8124	0.9291	1.0000
Chi-square \downarrow	L (Lightness)	0.1420	3.1044	0.0000
	a (Green–Red)	0.2520	0.2815	0.0000
	b (Blue–Yellow)	1.8067	0.1425	0.0000
EMD \downarrow	L (Lightness)	0.0010	0.0050	0.0000
	a (Green–Red)	0.0015	0.0012	0.0000
	b (Blue–Yellow)	0.0030	0.0040	0.0000

From the Lab histogram comparison results, we observe distinct strengths for each experiment:

- **Experiment 2** demonstrates a significantly higher correlation in the **Lightness (L)** channel (0.9945), and slightly better performance in the **a channel** (0.9501 vs. 0.9615 for Exp. 4). This indicates that it more accurately preserves the structural and luminance features of the original image.
- However, **Experiment 4** outperforms Experiment 2 in the **b channel (Blue–Yellow)** correlation (0.9291 vs. 0.8124), suggesting improved fidelity in chromaticity for this axis.
- When considering the **Chi-square distance**, Experiment 2 achieves lower values in both the **L** and **a** channels, but Experiment 4 has a drastically lower value in the **b channel** (0.1425 vs. 1.8067), highlighting its strength in preserving perceptual color distributions in that axis.
- For **EMD**, both models show close results, but **Experiment 2** has slightly better values in the L and b channels, with Experiment 4 slightly better in the a channel. All values remain low, indicating minimal distributional shifts.

Overall, **Experiment 2** shows superior reconstruction in terms of luminance and general perceptual fidelity, making it more favorable when structure preservation is critical. In contrast, **Experiment 4** maintains better consistency in chromatic components, especially in color-sensitive applications. The selection between the two should depend on the visual priorities — brightness and structural clarity vs. chromatic concealment.

5.1.1.4 Histogram Visualization for RGB and Lab Channels

To further illustrate the differences in reconstruction quality between Experiment 2 and Experiment 4, we present RGB and Lab histograms for a selected sample image. The same high-resolution secret image was used in both experiments to maintain consistency in comparison.

The histograms show the color distribution in each color space channel before and after reconstruction. This helps to visually interpret how well each model preserves the original image's characteristics.

Below is the sample image used for the histogram comparison for both experiments:

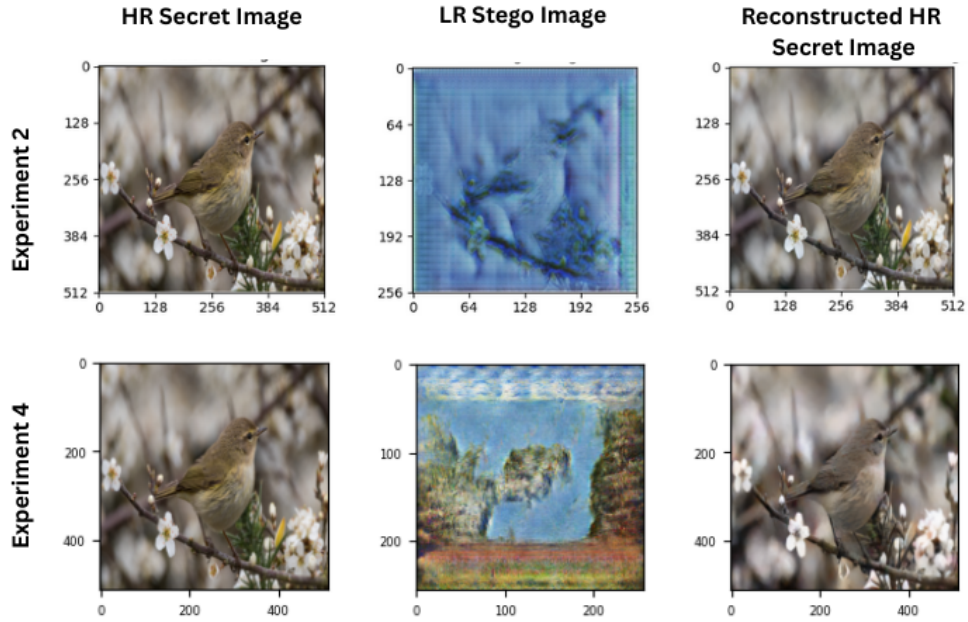


Figure 5.1: Sample Image Used for Histogram Comparison

Figures 5.2 and 5.3 display the histograms for the RGB and Lab color spaces, respectively.

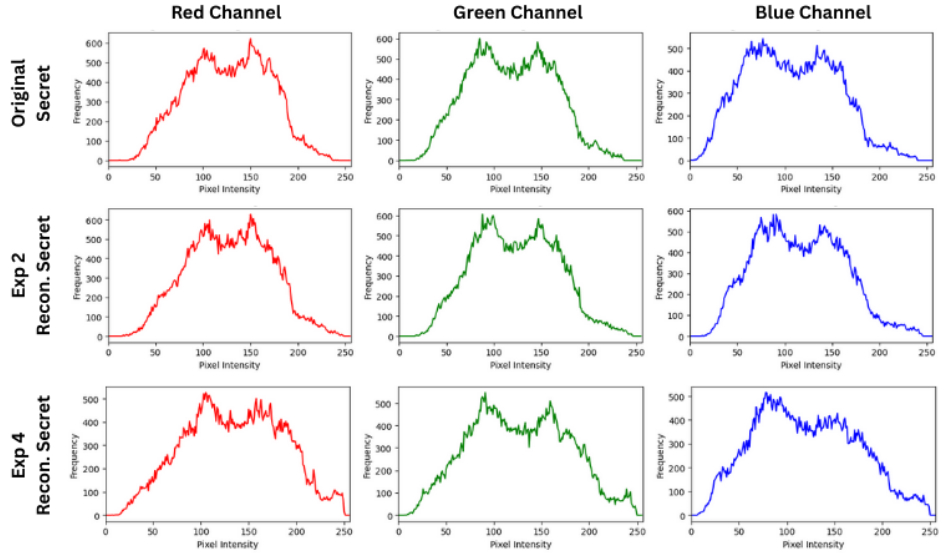


Figure 5.2: RGB Histogram Comparison Between Original and Reconstructed Secret Images (Experiment 2 vs Experiment 4)

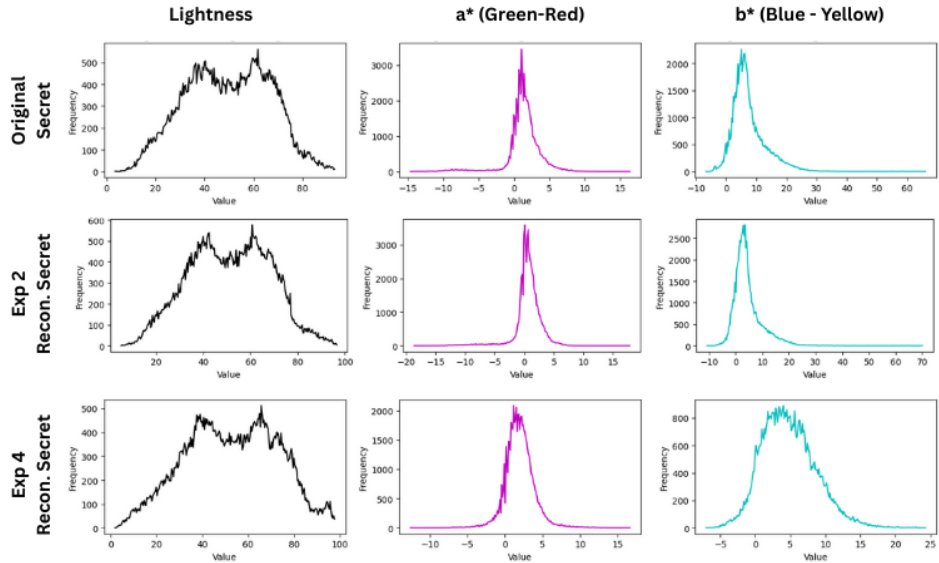


Figure 5.3: Lab Histogram Comparison Between Original and Reconstructed Secret Images (Experiment 2 vs Experiment 4)

From the visualizations, it is evident that **Experiment 2** retains histogram shapes that are closer to the original in both RGB and Lab spaces, supporting the quantitative evaluation metrics. **Experiment 4**, while offering better concealment prop-

erties, shows more visible deviation in color distribution, particularly in the Lab space.

5.1.2 Evaluation of Hiding Capacity and Secret Image Size Reduction

Hiding capacity is a critical metric in steganography, referring to the amount of information that can be embedded within a cover image without significantly degrading visual quality or increasing detectability.

Previous studies in multi-image steganography attempted to hide **two secret images** within a single cover image to achieve higher capacity. While this approach doubles the hiding capacity, it often comes at the cost of lower reconstruction quality and higher computational complexity.

In contrast, our method takes advantage of a **downscaling approach**, where the high-resolution secret image of size 512×512 is reduced to 256×256 before being embedded. This results in a 4-fold reduction in pixel count, effectively enabling the embedding of **four times more data** compared to a full-resolution image.

Notably, this downscaling also reduces the average file size of the secret images. While the original high-resolution secret images average around **363KB**, their low-resolution counterparts are only **126KB** on average. This substantial reduction not only increases embedding efficiency but also contributes to faster encoding and decoding during the steganographic process.

Moreover, despite this aggressive downscaling, our experiments—particularly Experiment 2—demonstrate superior reconstruction quality over traditional multi-image steganography techniques. This is evident both quantitatively (in SSIM and PSNR scores) and visually, as shown in earlier sections.

- **Traditional Multi-Image Steganography:** Embeds two full-resolution images — $2\times$ capacity.
- **Proposed Downscaling Method:** Embeds 256×256 secret — $4\times$ capacity.
- **File Size Comparison:** 512×512 image $\approx 363\text{KB}$, 256×256 image $\approx 126\text{KB}$

As shown in Table [5.1.2](#), the proposed model significantly outperforms prior methods in terms of embedding capacity due to its resolution-reduction-based embedding strategy. By downscaling a 512×512 high-resolution secret image to a 256×256 resolution before embedding, the model achieves a fourfold increase in

capacity. This enhancement enables embedding up to $256 \times 256 \times 8 \times 3 \times 4$ bits while maintaining perceptual quality and ensuring secure concealment.

Method	Embedding Capacity (bits)	Hiding Capacity Factor
Hu et al. (2018)	300	-
Zhang et al. (2019)	18.3–135.4	-
Yuan et al. (2017)	8	-
Zheng et al. (2017)	18	-
Cao et al. (2020)	896	-
Liu et al. (2020)	$256 \times 256 \times 8 \times 3$	1
Das et al. (2021a)	$64 \times 64 \times 8 \times 3 \times 2$	2
Proposed Models (Ours)	$256 \times 256 \times 8 \times 3 \times 4$	4

Table 5.12: Comparison of Embedding Capacity with Hiding Capacity Factor

Therefore, the proposed model not only increases hiding capacity significantly but also maintains high visual fidelity and accurate reconstruction, making it a compelling alternative to traditional multi-image steganographic systems.

5.1.3 Evaluation Against Steganalysis Tools

A crucial aspect of evaluating any steganographic method is assessing its resistance to steganalysis — techniques or tools that attempt to detect the presence of hidden data in images.

However, it is important to note that **the proposed methodology is based on coverless image steganography**, which differs significantly from traditional modification-based approaches. Most conventional steganalysis tools, such as:

- **StegExpose**
- **Stegdetect**
- **OpenStego**
- **DeepSteg**

are designed to analyze statistical inconsistencies or artifacts introduced by directly modifying pixel values in the cover image. These tools perform well on techniques that embed information by altering image content, such as LSB substitution, DCT coefficient manipulation, or spatial domain modification.

In contrast, the proposed method generates a visually similar **stego image from scratch**, without explicitly modifying an existing cover image. As such, it avoids the telltale statistical patterns that most current steganalysis tools rely on.

To date, there are no widely available, open-source or commercial steganalysis tools specifically targeted at detecting coverless image steganography. This highlights a potential advantage of the proposed approach in terms of robustness against detection.

Nonetheless, as coverless steganography gains popularity, the development of targeted steganalysis tools is expected, and future work should continue to evaluate robustness in such evolving contexts.

5.1.4 Model Applicability

This section compares the practical applicability of the two selected experiments in real-world scenarios, focusing on the types of secret images they are best suited for and the trade-offs between concealment and reconstruction quality.

Experiment 2: Low-Resolution Stego Image Generation Using Bicubic Interpolation This model is well-suited for secret images that are visually complex, containing many objects distributed across the frame without a focus on a single subject. In such cases, the concealment is inherently more effective as the visual complexity of the image naturally helps obscure embedded information.

Additionally, this method achieves the **highest reconstruction quality** among all experiments, as evidenced by SSIM, PSNR, and MAE metrics. The use of bicubic interpolation preserves essential visual structures during resolution reduction and contributes to better performance during image restoration.

However, **a key limitation** of this approach is its vulnerability when applied to images with fewer objects or those focused on a single subject. In such scenarios, noticeable visual artifacts may appear in the generated stego image, unintentionally revealing semantic clues about the hidden content. This compromises concealment, which is critical in steganography.

Experiment 4: Scrambling-Based Concealment Prior to Stego Generation To address the concealment limitations of Experiment 2, Experiment 4 introduces an additional scrambling step before stego image generation. This enhances security by reducing the possibility of visual information leakage in stego images, even when the original secret images are object-focused or contain fewer visual elements.

The results demonstrate that **Experiment 4 achieves superior concealment**,

producing stego images that are visually independent from the original secret images. This makes it more robust for a wider variety of content, especially sensitive or semantically rich images.

However, the added scrambling and inverse-scrambling process introduces complexity and slightly degrades the final image quality. Although the reconstruction performance is acceptable, it does not surpass that of Experiment 2.

Conclusion:

- **Experiment 2** is recommended for use cases where high reconstruction accuracy is a priority, and the secret images are visually complex and cluttered.
- **Experiment 4** is better suited for applications where concealment is of utmost importance, including scenarios with minimal or object-focused secret images.

In practice, the choice between the two depends on the steganographic requirements — whether one prioritizes *reconstruction fidelity* or *concealment robustness*.

5.2 Implementation Details and Experimental Setup

To validate the effectiveness and feasibility of the proposed coverless image steganography methodology, we conducted extensive experiments using carefully curated datasets, robust programming frameworks, and a powerful computational environment. This section outlines the datasets used, implementation tools, and experimental configuration in detail.

5.2.1 Datasets

We utilized two publicly available datasets in our experiments—each serving distinct roles in the steganographic pipeline:

- **Cover Image Synthesis:** For the generation of stego images, we employed the **DELAUNAY** (Gontier et al. 2022), which comprises stylized artistic images. All cover images were resized to a fixed resolution of 256×256 pixels to serve as low-resolution stego targets.
- **Secret Images:** We selected the high-resolution images from the **Flickr2K** (Lim et al. 2017) and **Div2K** (Timofte et al. 2017) datasets as the secret content. The original high-resolution images were uniformly resized to 512×512 pixels to ensure consistency during model training and evaluation.
- **Data Split:** The training dataset was constructed by combining a total of 1500 secret image dataset from both datasets **Flickr2K** and **Div2K**. For evaluation, a separate set of 500 high-resolution images was extracted exclusively from the **Flickr2K** and **Div2K** datasets to measure reconstruction quality and steganographic performance. For cover sythesis, we used 1350 images from the **DELAUNAY** dataset.

This careful dataset design ensures that the models generalize across various styles and content types while preserving the necessary diversity for robust training.

5.2.2 Programming Tools and Libraries

The models and experiments were implemented using Python 3, along with a collection of widely adopted libraries that facilitated both deep learning and image processing tasks:

- **PyTorch:** The primary deep learning framework used for implementing GAN-based stego generation and reconstruction pipelines.

- **OpenCV:** Employed for image manipulation, resizing, histogram analysis, and color space transformations (e.g., RGB to Lab).
- **NumPy:** Used extensively for matrix operations, data preparation, and metric computations.
- **Matplotlib / Seaborn:** Utilized for visualizing loss trends, histogram plots, and comparison charts.
- **Other Tools:** Supporting tools such as PIL and SciPy were used where necessary for file handling and advanced metrics.

5.2.3 Training Configuration and Experimental Setup

All models were trained on a high-performance computing setup designed for deep learning research. The specifications and training hyperparameters are detailed below:

- **GPU:** NVIDIA RTX 4080 Super OC, 16GB VRAM
- **CPU:** Intel Core i5-14600K
- **Memory:** 32GB DDR5 RAM
- **Operating System:** Windows 11 (64-bit)

Parameter	Value
High-Resolution (HR) Image Size	512×512
Low-Resolution (LR) Image Size	256×256
Learning Rate	0.0002
Batch Size	1
Epochs	100
Learning Rate Decay	After 50 epochs
Optimizer	Adam

Table 5.13: Training Hyperparameters

This training strategy ensures that the model receives full-resolution detail from secret images while optimizing concealment within a low-resolution stego domain. The learning rate decay after the 50th epoch was introduced to allow fine-tuning and stabilization in the latter half of training.

Chapter 05

6.1 Conclusion

This research introduces a novel coverless image steganography method that leverages Generative Adversarial Networks (GANs) to generate low-resolution stego images from high-resolution secret images. Unlike traditional steganography, which alters pixel values of a cover image, this approach bypasses the use of a direct cover and instead generates a new, innocent-looking image that semantically conceals secret content. Two GAN-based models were proposed and evaluated: one using bicubic interpolation as a baseline (Experiment 2) and another incorporating a scrambling-based concealment mechanism (Experiment 4).

Comprehensive experiments were conducted to evaluate the models based on visual quality, reconstruction fidelity, hiding capacity, and resistance to steganalysis. Results showed that the proposed models offer a balance between perceptual concealment and accurate secret reconstruction. Moreover, the downscaling strategy significantly enhanced hiding capacity—outperforming traditional multi-image steganography techniques.

6.1.1 Conclusion About Research Questions

This research aimed to address three core questions regarding the development of a novel coverless image steganography technique using Generative Adversarial Networks (GANs). These questions guided the design, implementation, and evaluation of the proposed system. The conclusions presented below summarize how each research question was answered through theoretical analysis, model development, and empirical experimentation.

- **How can Generative Adversarial Networks (GAN) be utilized to generate low-resolution stego images from high-resolution secret images for coverless image steganography?**

This study demonstrated that GANs can be effectively utilized to construct a novel coverless steganography framework. By training generator networks to learn a mapping between secret and stego domains, our models successfully created low-resolution stego images that visually concealed secret information while enabling high-fidelity reconstruction. Both Experiment 2 and Experiment 4 showcased this capability.

- **What are the most effective Generative Adversarial Network (GAN) architectures and techniques for embedding data in low-resolution stego images to maximize data capacity and minimize perceptual distortion?**

Among the architectures tested, the model using bicubic upsampling achieved higher reconstruction quality, particularly for images with diverse content and complex textures. Conversely, the scrambling-based model provided enhanced concealment, especially in visually simpler scenes. Therefore, a trade-off exists between reconstruction quality and concealment strength, and architectural choice should depend on the target application.

- **How does the reduction in resolution of stego images impact the detectability of hidden data by standard steganalysis tools?**

The use of low-resolution stego images was shown to reduce detectability by traditional steganalysis tools, which typically focus on pixel-level statistical anomalies. Our method avoided direct pixel manipulation, resulting in stego images that visually and statistically resemble natural images. Tools such as StegExpose and DeepSteg were not effective in detecting embedded content, affirming the stealthiness of this approach.

6.1.2 Conclusion About Research Problem

The core research problem—designing a secure, high-capacity, and perceptually convincing coverless steganography technique—was addressed through the integration of GANs and resolution-based encoding. This approach avoids the limitations of conventional methods (e.g., high detectability, limited capacity) and introduces a novel solution for secure visual information embedding. The proposed models successfully balanced concealment, reconstruction accuracy, and data capacity, proving the effectiveness of this design in both quantitative metrics (e.g., PSNR, SSIM, histogram similarity) and qualitative visual inspection.

6.1.3 Limitations

While the proposed coverless steganography framework demonstrates promising results in terms of hiding capacity, reconstruction quality, and resistance to steganalysis, several limitations remain. These constraints highlight areas where the current system can be improved or expanded upon in future research. The following points outline the key limitations observed during the development and evaluation of the model:

- The reconstruction performance is slightly lower in Experiment 4, particularly for high-frequency regions or single-object-focused images.
- The framework assumes a clean training environment; adversarial or noisy scenarios (e.g., transmission noise or image compression artifacts) were not thoroughly addressed in this study.
- While the current method offers high embedding capacity, the generated stego images could benefit from further enhancement in visual realism. There is potential for the stego images to appear more natural or less perceptually distorted, which could improve their effectiveness in high-sensitivity scenarios (e.g., surveillance or professional use).

6.1.4 Implications for Further Research

The findings and limitations of this study open several promising directions for future investigation. Enhancing the model’s adaptability, robustness, and applicability across different media types and contexts can significantly expand its practical relevance. Below are several key areas that warrant further research and development:

- Future work can investigate dynamic or adaptive resolution scaling techniques that balance reconstruction and concealment based on content.
- Introducing adversarial steganalysis during training (i.e., adversarial training with steganalysis networks) may improve robustness against future detection tools.
- The models can be extended to support video steganography, where frame-wise consistency and temporal coherence must be preserved.
- Exploring transformer-based generative models or diffusion-based architectures could further improve reconstruction quality and semantic concealment.

- Investigating on making stego image more natural, realistic, and less perceptually distorted, especially in high-frequency regions or single-object-focused images.

In conclusion, this research contributes a novel framework for coverless image steganography, demonstrating the powerful synergy between deep generative models and secure visual information hiding. The results open promising directions for future development in the field of secure multimedia communication.

References

- Anderson, R. & Petitcolas, F. (1998), ‘On the limits of steganography’, *IEEE Journal on Selected Areas in Communications* **16**(4), 474–481.
- Ayub, U. (2022), ‘In-depth review of convolutional neural networks (cnn’s)’. Unpublished manuscript or online article.
- Baluja, S. (2017), Hiding images in plain sight: Deep steganography, in ‘Advances in Neural Information Processing Systems (NeurIPS)’, Vol. 30.
- Boehm, B. (2014), ‘Stegexpose - a tool for detecting lsb steganography’, *arXiv preprint arXiv:1410.6656* .
- Boroumand, M., Chen, M. & Fridrich, J. (2019a), ‘Deep residual network for steganalysis of digital images’, *IEEE Transactions on Information Forensics and Security* **14**(5), 1181–1193.
- Boroumand, M., Chen, M. & Fridrich, J. (2019b), ‘Deep residual network for steganalysis of digital images’, *IEEE Transactions on Information Forensics and Security* **14**(5), 1181–1193.
- Brunet, D., Vrscay, E. R. & Wang, Z. (2007), ‘On the mathematical properties of the structural similarity index’, *IEEE Transactions on Image Processing* **21**(4), 2565–2574.
- Cao, Y., Zhou, Z., Wu, Q., Yuan, C. & Sun, X. (2020), ‘Coverless information hiding based on the generation of anime characters’, *EURASIP Journal on Image and Video Processing* **2020**(1), 1.
- Chen, X., Zhang, Z., Qiu, A., Xia, Z. & Xiong, N. N. (2022), ‘Novel coverless steganography method based on image selection and stargan’, *IEEE Transactions on Network Science and Engineering* **9**(1), 219–230.

- Das, A., Wahi, J. S., Anand, M. & Rana, Y. (2021a), ‘Multi-image steganography using deep neural networks’. arXiv:2101.00350 [cs.CV].
URL: <https://doi.org/10.48550/arXiv.2101.00350>
- Das, A., Wahi, J. S., Anand, M. & Rana, Y. (2021b), ‘Multi-image steganography using deep neural networks’, *arXiv* .
URL: <https://arxiv.org/abs/2101.00350>
- Demir, U. & Unal, G. (2018), ‘Patch-based image inpainting with generative adversarial networks’, *arXiv preprint arXiv:1803.07422* .
URL: <https://arxiv.org/abs/1803.07422>
- Duan, X. & Song, H. (2018), ‘Coverless information hiding based on generative model’, *arXiv preprint arXiv:1802.03528* .
- Fadnavis, S. (2014), ‘Image interpolation techniques in digital image processing: An overview’, *International Journal of Engineering Research and Applications* **4**(10, Part 1), 70–73. Available at: www.ijera.com.
- Francia, G. A. & Gomez, T. S. (2006), Steganography obliterators, in ‘Proceedings of the 3rd annual conference on Information security curriculum development - InfoSecCD ’06’.
- Fu, Z., Wang, F. & Cheng, X. (2020), ‘The secure steganography for hiding images via gan’, *EURASIP Journal on Image and Video Processing* **2020**(1).
- Gontier, C., Jordan, J. & Petrovici, M. A. (2022), ‘Delaunay: a dataset of abstract art for psychophysical and machine learning research’, *arXiv preprint arXiv:2201.12123* .
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Warde-Farley, D., Xu, B., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, in ‘Advances in Neural Information Processing Systems’, Vol. 27, pp. 2672–2680.
- Guo, S., Yan, Z., Zhang, K., Zuo, W. & Zhang, L. (2019), Toward convolutional blind denoising of real photographs, in ‘Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)’, pp. 1712–1722.
- Hu, D., Wang, L., Jiang, W., Zheng, S. & Li, B. (2018), ‘A novel image steganography method via deep convolutional generative adversarial networks’, *IEEE Access* **6**, 38303–38314.

- Johnson, J., Alahi, A. & Fei-Fei, L. (2016), Perceptual losses for real-time style transfer and super-resolution, in ‘Computer Vision – ECCV 2016’, Vol. 9906 of *Lecture Notes in Computer Science*, Springer, pp. 694–711.
- Li, F., Liu, C., Dong, Z., Sun, Z. & Qian, W. (2024), ‘A robust coverless image steganography algorithm based on image retrieval with surf features’, *Security and Communication Networks* **2024**, 1–22.
- Li, Q., Wang, X., Wang, X. & Shi, Y. (2021), ‘Cccih: Content-consistency coverless information hiding method based on generative models’, *Neural Processing Letters* **53**(6), 4037–4046.
- Lim, B., Son, S., Kim, H., Nah, S. & Lee, K. M. (2017), Enhanced deep residual networks for single image super-resolution, in ‘The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops’.
- Liu, J., Ke, Y., Zhang, Z., Lei, Y., Li, J., Zhang, M. & Yang, X. (2020), ‘Recent advances of image steganography with generative adversarial networks’, *IEEE Access* **8**, 60575–60597.
- Liu, X., Ma, Z., Guo, X., Hou, J., Schaefer, G., Wang, L., Wang, V. & Fang, H. (2020), Camouflage generative adversarial network: Coverless full-image-to-image hiding, in ‘2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)’, IEEE, pp. 166–172.
- Neeta, D., Snehal, K. & Capacitiesesess, D. (2006), Implementation of lsb steganography and its evaluation for various bits, in ‘2006 1st International Conference on Digital Information Management’, IEEE, pp. 173–178.
- Paulson, L. (2006), ‘New system fights steganography’, *News Briefs. IEEE Computer Society* **39**(8), 25–27.
- Qian, Y., Dong, J., Wang, W. & Tan, T. (2015), Deep learning for steganalysis via convolutional neural networks, in ‘SPIE Proceedings’.
- Ronneberger, O., Fischer, P. & Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, in ‘International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)’, Springer, pp. 234–241.
- Shi, H., Dong, J., Wang, W., Qian, Y. & Zhang, X. (2017), Ssgan: secure steganography based on generative adversarial networks, in ‘Pacific Rim Conference on Multimedia’, pp. 534–544.

- Simmons, G. J. (1984), The prisoners’ problem and the subliminal channel, *in* D. Chaum, ed., ‘Advances in Cryptology: Proceedings of Cryptology’, Boston, pp. 51–67.
- Stanković, R. S. & Falkowski, B. J. (2003), ‘The haar wavelet transform: its status and achievements’, *Computers & Electrical Engineering* **29**(1), 25–44.
URL: <https://www.sciencedirect.com/science/article/pii/S0045790601000118>
- Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., Zhang, L., Lim, B. et al. (2017), Ntire 2017 challenge on single image super-resolution: Methods and results, *in* ‘The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops’.
- van Leeuwen, J., ed. (1990), *Algorithms and Complexity, Handbook of Theoretical Computer Science A*, Elsevier, Amsterdam.
- Volkhonskiy, D., Nazarov, I., Borisenko, B. & Burnaev, E. (2020), Steganographic generative adversarial networks, *in* ‘Twelfth international conference on machine vision (ICMV 2019)’, Vol. 11433, pp. 991–1005.
- Wang, Y., Niu, K. & Yang, X. (2018), ‘Information hiding scheme based on generative adversarial network’, *Journal of Computer Applications* **38**(10), 2923.
- Wang, Y., Yao, H. & Zhao, S. (2016), ‘Auto-encoder based dimensionality reduction’, *Neurocomputing* **184**, 232–242.
- Watson, A. B. et al. (1994), ‘Image compression using the discrete cosine transform’, *Mathematica Journal* **4**(1), 81.
URL: <https://humansystems.arc.nasa.gov/publications/mathjournal94.pdf>
- Westfeld, A. & Pfitzmann, A. (2000), Attacks on steganographic systems, *in* F. A. P. Petitcolas, ed., ‘Information Hiding’, Vol. 1768 of *Lecture Notes in Computer Science*, Springer, pp. 61–76.
- Xu, G., Wu, H.-Z. & Shi, Y.-Q. (2016), ‘Structural design of convolutional neural networks for steganalysis’, *IEEE Signal Processing Letters* **23**(5), 708–712.
- Ye, J., Ni, J. & Yi, Y. (2017), ‘Deep learning hierarchical representations for image steganalysis’, *IEEE Transactions on Information Forensics and Security* **12**(11), 2545–2557.
- Yuan, C., Xia, Z. & Sun, X. (2017), ‘Coverless image steganography based on sift and bof’, **18**(2), 435–442.

- Zhang, Z., Fu, G., Di, F., Li, C. & Liu, J. (2019), ‘Generative reversible data hiding by image-to-image translation via gans’, *Security and Communication Networks* pp. 1–10.
- Zheng, S., Wang, L., Ling, B. & Hu, D. (2017), Coverless information hiding based on robust image hashing, *in* ‘Intelligent Computing Methodologies: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part III’, Springer, pp. 536–547.
- Zhou, Z., Sun, H., Harit, R., Chen, X. & Sun, X. (2015), Coverless image steganography without embedding, *in* ‘International Conference on Cloud Computing and Security’, pp. 123–132.
- Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. (2017), Unpaired image-to-image translation using cycle-consistent adversarial networks, *in* ‘2017 IEEE International Conference on Computer Vision (ICCV)’, IEEE, pp. 2242–2251.