



People's Democratic  
Republic of Algeria  
Ministry of Higher Education  
and Scientific Research



## Bouira University

Faculty of Exact Sciences  
Department of Computer Science

## Master Thesis

---

# Robust Steganography Techniques for Data Tampering Resilience

---

**Submitted by:**

Abid Akram  
Tahraoui Mustapha

**Supervised by:**

Dr. Benaissi Sellami

Submitted in partial fulfillment of the requirements for the degree of  
**Master in Computer Science**

Academic Year: 2025 – 2026

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	General Background . . . . .	6
1.2	Problem Statement . . . . .	6
1.3	Research Objectives . . . . .	7
1.4	Scope and Contributions . . . . .	7
1.5	Thesis Organization . . . . .	8
<b>2</b>	<b>Literature Review</b>	<b>9</b>
2.1	Fundamentals of Image Steganography . . . . .	9
2.1.1	Imperceptibility, Capacity, and Robustness Trade-off . . . . .	9
2.2	Spatial Domain Techniques . . . . .	10
2.2.1	Transform Domain Techniques . . . . .	10
2.2.2	Discrete Cosine Transform (DCT)-Based Steganography . . . . .	10
2.2.3	Discrete Wavelet Transform (DWT)-Based Steganography . . . . .	15
2.2.4	Discrete Fourier Transform (DFT)-Based Steganography . . . . .	17
2.3	Robustness-Oriented Approaches . . . . .	23
2.3.1	Error Correction Coding for Steganographic Robustness . . . . .	23
2.3.2	The Synchronization Problem . . . . .	24
2.3.3	Spread-Spectrum Steganography . . . . .	25
2.3.4	Adaptive Embedding and Perceptual Masking . . . . .	26
2.3.5	Fragmentation, Redundancy, and Distributed Embedding . . . . .	26
2.3.6	Robust Steganography vs. Digital Watermarking . . . . .	27
2.3.7	Steganalysis and the Detectability–Robustness Trade-off . . . . .	28
2.3.8	Deep Learning–Based Robust Steganography . . . . .	28
2.3.9	Discussion and Research Gap . . . . .	29
2.4	Attacks and Image Degradation Models . . . . .	30
2.4.1	Signal Processing Attacks . . . . .	30
2.4.2	Geometric Attacks . . . . .	31
2.4.3	Impact of Attacks on Steganographic Robustness . . . . .	32
2.5	Discussion and Research Gap . . . . .	32
2.6	Recent Advances in Deep Learning-Based Steganography . . . . .	33
2.6.1	The Shift Toward End-to-End Learned Steganography . . . . .	33
2.6.2	HiDDeN: The Foundational Framework (2018) . . . . .	34
2.6.3	SteganoGAN: High-Capacity GAN-Based Steganography (2019) . . . . .	35
2.6.4	ReDMark: Robustness-First Deep Steganography (2021) . . . . .	36
2.6.5	Hiding Images in Images: High-Capacity Deep Concealment . . . . .	36
2.6.6	Robust Deep Steganography Against Social Network Processing . . . . .	37
2.6.7	Attention Mechanisms and Semantic-Aware Embedding . . . . .	37
2.6.8	Hybrid Transform-Network Approaches . . . . .	37
2.6.9	Deep Learning Steganalysis and Its Impact on Robust Design . . . . .	38
2.6.10	Summary Table and Comparative Analysis . . . . .	38
2.6.11	Why This Thesis Adopts a Classical Framework . . . . .	38

2.7	Evaluation Criteria in Steganographic Systems . . . . .	40
2.7.1	Imperceptibility Metrics . . . . .	40
2.7.2	Robustness Metrics . . . . .	43
2.7.3	Partial Recovery Rate (PRR) . . . . .	44
2.7.4	Unified View of Evaluation Criteria . . . . .	45
<b>3</b>	<b>Proposed Self-Recovering Steganography Framework</b>	<b>46</b>
3.1	Design Motivation and Objectives . . . . .	46
3.2	High-Level System Overview . . . . .	46
3.2.1	Sender-Side Processing Pipeline . . . . .	46
3.2.2	Receiver-Side Processing Pipeline . . . . .	47
3.3	Structured Payload Representation . . . . .	47
3.3.1	Payload Components . . . . .	47
3.3.2	Inter-Fragment Relationships . . . . .	47
3.4	Fragmentation Strategy . . . . .	47
3.4.1	Fragment Size and Granularity . . . . .	47
3.4.2	Redundancy and Distribution Policy . . . . .	48
3.5	Multi-Domain Embedding Strategy . . . . .	48
3.5.1	Spatial Domain Embedding . . . . .	48
3.5.2	Frequency Domain Embedding . . . . .	48
3.5.3	Multi-Scale and Redundant Embedding . . . . .	48
3.5.4	Color Channel Utilization . . . . .	48
3.6	Embedding Control and Parameter Selection . . . . .	48
3.6.1	Key-Based Fragment Placement . . . . .	48
3.6.2	Embedding Strength Adaptation . . . . .	48
3.7	Damage-Aware Extraction Process . . . . .	49
3.7.1	Fragment Detection and Validation . . . . .	49
3.7.2	Confidence Scoring Mechanism . . . . .	49
3.7.3	Adaptive Fragment Selection . . . . .	49
3.8	Message Reconstruction and Self-Recovery . . . . .	49
3.8.1	Partial Reconstruction Strategy . . . . .	49
3.8.2	Failure Conditions . . . . .	49
3.9	Computational Complexity and Practical Considerations . . . . .	49
3.9.1	Complexity Analysis . . . . .	49
3.9.2	Implementation Constraints . . . . .	49
3.10	Summary of the Proposed Framework . . . . .	50
<b>4</b>	<b>Implementation smdljkf and Experimental Setup</b>	<b>51</b>
4.1	Development Environment . . . . .	51
4.2	Dataset Description . . . . .	51
4.3	Attack Simulation . . . . .	51
4.4	Evaluation Metrics . . . . .	51
4.4.1	Imperceptibility Metrics . . . . .	51
4.4.2	Robustness Metrics . . . . .	51
<b>5</b>	<b>Results and Discussion</b>	<b>52</b>
5.1	Imperceptibility Evaluation . . . . .	52
5.2	Robustness Under Image Degradation . . . . .	52
5.3	Robustness Under Partial Data Loss . . . . .	52
5.4	Comparative Analysis . . . . .	52
5.5	Discussion of Results . . . . .	52

<b>6 Conclusion and Future Work</b>	<b>53</b>
6.1 Summary of Findings . . . . .	53
6.2 Limitations . . . . .	53
6.3 Future Research Directions . . . . .	53

# List of Figures

2.1	Steganography Trilemma: Three Competing Goals . . . . .	10
2.2	DCT coefficient distribution showing low-, mid-, and high-frequency components in an $8 \times 8$ block . . . . .	11
2.3	DCT deivision image in an $8 \times 8$ blocks . . . . .	12
2.4	DC and AC coefficient on DCT Transformation . . . . .	13
2.5	Block-based DCT embedding process illustrating transformation, coefficient selection, and inverse transformation. . . . .	14
2.6	One-level and multi-level DWT decomposition showing LL, LH, HL, and HH sub-bands. . . . .	16
2.7	DFT-based embedding process illustrating frequency selection, magnitude modification, and inverse transformation . . . . .	16
2.8	Generic encoder–decoder architecture for deep steganography. The noise layer simulates signal processing and geometric attacks during training, forcing the encoder to learn embeddings that survive realistic degradation. An optional discriminator provides an adversarial loss $\mathcal{L}_{\text{adv}}$ to penalise statistically detectable stego-images. . . . .	34
2.9	Illustrative relationship between embedding strength $\alpha$ and the two imperceptibility metrics PSNR and SSIM (scaled by 50). Both metrics decrease monotonically with increasing embedding strength. PSNR degrades rapidly even at moderate strengths, while SSIM is more tolerant of small-amplitude, high-frequency embedding signals that preserve structural content. This highlights the complementary nature of the two metrics. . . . .	42
2.10	Relationship between the three steganographic objectives and the quantitative evaluation metrics used in this thesis. The trade-offs between objectives are captured by tracking all metrics simultaneously across experimental conditions. PRR (Partial Recovery Rate) is a metric introduced in this work to assess the unique partial-recovery capability of the proposed framework. . . . .	45

# Contents

# Chapter 1

## Introduction

### 1.1 General Background

In contrast to cryptography, which focuses on protecting the content of a message by transforming it into an unreadable form, steganography aims to conceal the very existence of communication. While digital watermarking shares similarities with steganography, it is primarily designed for ownership verification and copyright protection rather than covert communication. Robust steganography lies at the intersection of these domains, seeking to maintain hidden communication while ensuring resistance to intentional or unintentional image modifications.

In real-world digital environments, images rarely remain unchanged after distribution. Common platforms such as social media networks, cloud storage services, and messaging applications routinely apply operations including lossy compression, resizing, filtering, and format conversion. These automatic transformations pose a significant threat to traditional steganographic techniques, which are often designed under ideal transmission assumptions. As a result, the development of steganographic systems capable of surviving such uncontrolled and hostile processing environments has become an essential research challenge.

Steganography is the art and science of invisible communication, achieved by concealing information within other digital content in such a way that the existence of the hidden message remains undetectable. The term steganography is derived from the Greek words *stegos*, meaning “cover,” and *graphia*, meaning “writing,” and is commonly interpreted as “covered writing.”[1]. In the context of image steganography, secret information is embedded exclusively within digital images, which serve as the cover medium for hidden communication.

### 1.2 Problem Statement

From the analysis above, the core challenges addressed in this work can be summarized as follows:

- Existing steganographic methods exhibit high fragility when subjected to common signal processing operations such as compression, noise addition, and filtering, resulting in partial or complete loss of embedded information.
- Geometric transformations, including cropping and resizing, introduce spatial desynchronization between the embedding and extraction processes, severely limiting reliable data recovery.
- Most current approaches prioritize imperceptibility while neglecting the survivability of the hidden payload under compound or severe distortions.
- The majority of embedding strategies focus on protecting embedding locations rather than enabling resilience or reconstruction of the hidden message itself.

These limitations highlight the need for a steganographic framework that shifts the design focus from invisibility-centered embedding toward payload-oriented survivability, enabling partial recovery and reliable communication in hostile and uncontrolled digital environments.

## 1.3 Research Objectives

The primary objective of this research is to address the limitations of conventional image steganography techniques by prioritizing the survivability of hidden data under adverse and uncontrolled image degradation. Rather than focusing exclusively on high-capacity or imperceptible embedding, this work aims to enhance the resilience of the embedded payload against both signal processing and geometric distortions.

The specific objectives of this thesis are as follows:

- To design a robust steganographic framework that improves the survivability of hidden information under severe and compound image degradations.
- To structure the secret payload into interconnected fragments that enable partial data recovery when segments of the stego-image are lost or corrupted.
- To distribute embedded message fragments across stable image regions and multiple transform domains in order to reduce the impact of localized tampering.
- To develop an extraction and reconstruction strategy capable of recovering usable information even in the presence of spatial misalignment and partial data loss.
- To evaluate the proposed framework using standard image quality and robustness metrics, including PSNR [2] and Bit Error Rate (BER) [3], under realistic attack scenarios.

## 1.4 Scope and Contributions

This research focuses on robust data hiding in digital images, with particular emphasis on improving the survivability of hidden information under common image degradation and tampering scenarios. The scope of this work is limited to still images and does not address steganography in video, audio, or real-time streaming media. Furthermore, the proposed framework does not aim to maximize payload capacity, but instead prioritizes reliable data recovery under adverse conditions.

The study assumes that image degradation may occur due to both unintentional processing (such as compression and resizing) and intentional tampering (such as cropping or localized modification). However, the framework does not guarantee successful recovery under all extreme or adversarial scenarios, particularly in cases of complete image destruction or aggressive multi-stage attacks.

The main contributions of this thesis can be summarized as follows:

- Proposing a robustness-oriented steganographic framework that emphasizes payload survivability rather than solely protecting embedding locations.
- Introducing a structured message fragmentation strategy that enables partial reconstruction of the hidden data when portions of the stego-image are lost or corrupted.
- Designing a multi-domain embedding strategy that distributes message fragments across different image regions to reduce sensitivity to localized distortions.
- Developing an extraction and reconstruction mechanism capable of recovering meaningful information under spatial misalignment and partial data loss.
- Providing an experimental evaluation of the proposed framework under realistic signal processing and geometric attack scenarios.

## 1.5 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents a comprehensive review of existing steganographic techniques, with particular emphasis on robustness-oriented approaches and their limitations under image degradation and tampering. Chapter 3 introduces the proposed robust steganography framework, detailing its design principles, message structuring strategy, and embedding and extraction mechanisms. Chapter 4 describes the implementation details, experimental setup, datasets, attack models, and evaluation metrics used in this study. Chapter 5 presents and discusses the experimental results, including imperceptibility and robustness analyses as well as comparative evaluations with classical methods. Finally, Chapter 6 concludes the thesis by summarizing the main findings, discussing limitations, and outlining directions for future research.

# Chapter 2

## Literature Review

### 2.1 Fundamentals of Image Steganography

Image steganography refers to techniques that embed secret information within digital images while preserving the visual appearance of the carrier. In contrast to introductory definitions presented earlier, this chapter focuses on how steganographic systems are modeled, evaluated, and constrained in practical scenarios. A widely adopted theoretical model is the prisoner's problem, in which the security of a steganographic system depends on the inability of an adversary to statistically distinguish between cover images and stego-images [4]. This model emphasizes that effective steganography must balance invisibility with functional robustness under realistic transmission conditions.

#### 2.1.1 Imperceptibility, Capacity, and Robustness Trade-off

The efficacy of the image steganography system is inevitably limited by the trade-off between the following two factors: between three key performance criteria: imperceptibility, payload capacity, and robustness. These Three properties are necessarily interconnected so that enhancement in one area will result in improvement in other areas as well. decline in at least one of the others. This pattern is often represented in a triangular relationship: This represents a trade-off and is one of the main challenges in designing a steganographic system [5].

- **Imperceptibility** refers to the degree to which a stego-image remains visually and statistically indistinguishable from the corresponding cover image. Imperceptibility is a crucial factor in steganography to avoid being detected either by human observers or statistical steganalysis algorithms. Methods which Aggressively embedded data could introduce artifacts that affect either the statistical distribution or signal quality. For example, data that has risk of detection [6].
- **Payload capacity** denotes the amount of secret information that can be embedded within a cover image. While higher capacity improves communication efficiency, it typically requires stronger or more frequent modifications to the image content, which negatively impacts imperceptibility and increases vulnerability to image processing operations. Consequently, high-capacity embedding schemes often sacrifice robustness in order to maximize payload [7].
- **Robustness** describes the ability of the embedded information to survive image degradation caused by signal processing operations or intentional attacks. Robust steganographic schemes are designed to withstand distortions such as compression, noise addition, filtering, and geometric transformations. Achieving robustness usually requires redundancy, structured embedding, or the use of stable image components, which in turn reduces the effective payload capacity [8].

In the context of robust steganography, robustness is frequently prioritized over capacity, particularly in applications where data recovery is more critical than transmission efficiency. As a result,

Image steganography is the practice of concealing secret information within an ordinary digital image. Its effectiveness is not measured by a single metric but by a delicate balance between three competing goals. Improving one of these properties almost always weakens at least one of the others.

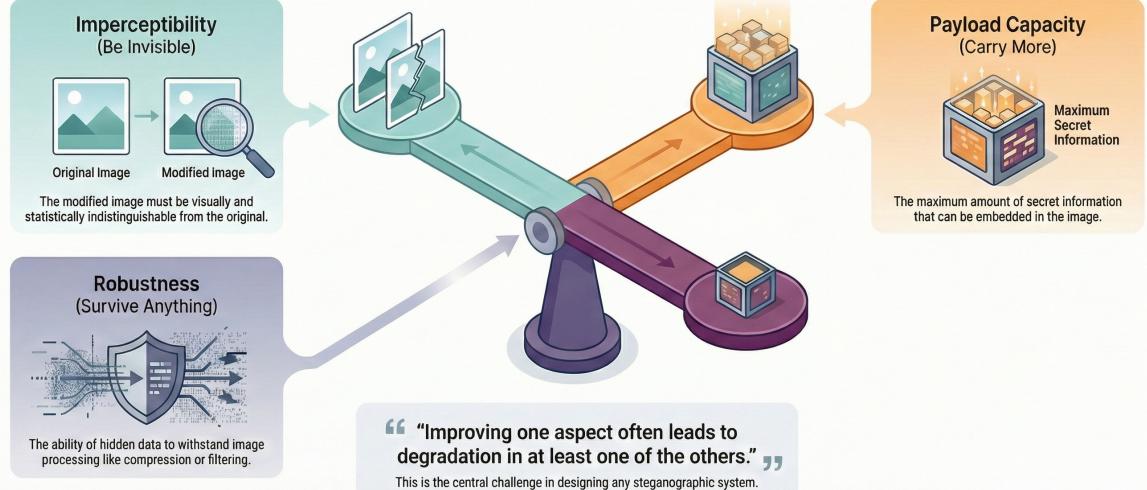


Figure 2.1: Steganography Trilemma: Three Competing Goals

many robust systems intentionally operate at lower payload rates to ensure reliable extraction under adverse conditions. Balancing these three competing objectives remains a key research challenge and motivates the exploration of resilience-oriented steganographic frameworks.

## 2.2 Spatial Domain Techniques

Spatial domain approaches embed the secret by directly modifying the intensities of pixels in the cover image. Among the methods, one can distinguish LSB substitution and its variants are the most studied families due to their simplicity and low computational complexity. High embedding capacity[7]. In LSB-based methods, the least significant bits of pixel values are changed to convey secret information, leading to less than ideally perceivable distortion.

Spatial domain techniques, despite all their advantages, suffer from some serious drawbacks in terms of robustness. Since the embedding procedure acts directly on pixel values, the hidden data is highly susceptible under common image processing manipulations. Even light lossy compression, noise addition, Filtering irreversibly destroys some fractional bits of the least significant bits, and the result can be a partial or complete loss of the embedded message[6]. Hence, most spatial domain methods are unsuitable for applications that require robustness against image degradation or tampering as a main constraint.

### 2.2.1 Transform Domain Techniques

According to the Discrete Cosine Transform (DCT), the DCT is an example of a method of digital image steganography that embeds information into the transform domain of a digital image, instead of modifying the pixel intensity values directly. The DCT takes advantage of the ability of the human visual system to be less sensitive to different spectral components in the image, particularly when working in the frequency or multi-resolution space[8].

### 2.2.2 Discrete Cosine Transform (DCT)-Based Steganography

The Discrete Cosine Transform (DCT) is a fundamental transform-domain technique widely used in image steganography due to its strong connection with the JPEG compression standard. Unlike spatial-domain approaches that directly modify pixel intensity values, DCT-based steganography embeds secret information within the frequency coefficients of an image. This allows data hiding

to exploit the characteristics of the human visual system, which is less sensitive to modifications in certain frequency components [5, 8].

## Overview of the DCT Transformation

The DCT converts spatial image data into a frequency-domain representation. In practice, a grayscale or luminance image is first divided into non-overlapping blocks of size  $8 \times 8$  pixels. Each block is then transformed independently using the DCT, resulting in 64 coefficients that represent different spatial frequency components.

The resulting coefficients can be categorized as:

- **Low-frequency coefficients:** Represent the average intensity and coarse image structures.
- **Mid-frequency coefficients:** Represent edges and moderate texture details.
- **High-frequency coefficients:** Represent fine details and noise-like components.

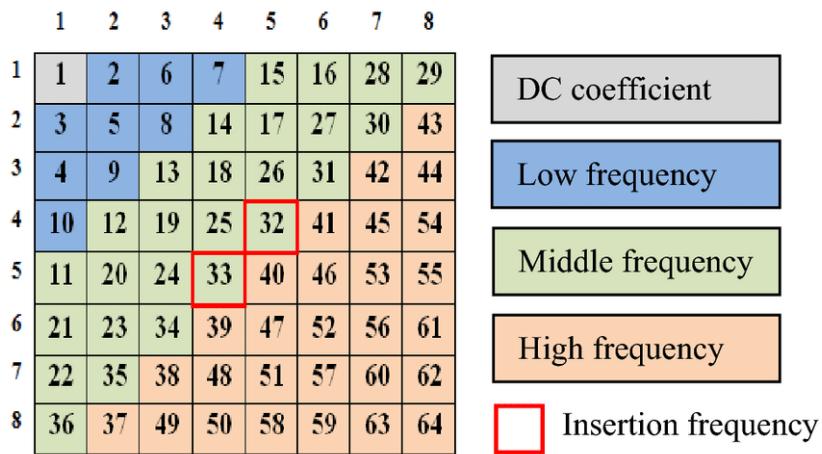


Figure 2.2: DCT coefficient distribution showing low-, mid-, and high-frequency components in an  $8 \times 8$  block

## Embedding Process in DCT-Based Steganography

The embedding procedure in DCT-based steganography typically follows a structured sequence of steps:

1. **Color Space Conversion:** For color images, the RGB image is commonly converted into a luminance-chrominance color space (such as YCbCr). Data embedding is primarily performed in the luminance (Y) channel, as it carries the most perceptual information.

**The YCbCr Components** The YCbCr space represents an image through three distinct components:

- **Y (Luminance):** Represents the brightness information of the image.
- **Cb (Chroma Blue):** Represents the difference between the blue component and a reference value.
- **Cr (Chroma Red):** Represents the difference between the red component and a reference value.

**Mathematical Transformation** The conversion from the digital RGB space to YCbCr is defined by a linear transformation. According to the ITU-R BT.601 standard, the transformation matrix for 8-bit digital signals is expressed as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

In a more simplified normalized form, the equations can be written as:

$$Y = 0.299R + 0.587G + 0.114B \quad (2.2)$$

$$Cb = -0.1687R - 0.3313G + 0.5B + 128 \quad (2.3)$$

$$Cr = 0.5R - 0.4187G - 0.0813B + 128 \quad (2.4)$$

**Significance for Robust Steganography** By utilizing the YCbCr space, the steganographic algorithm can selectively embed data into the chrominance channels ( $Cb$  or  $Cr$ ). Since the Human Visual System (HVS) is less sensitive to these channels, the hidden data can better withstand degradation such as lossy JPEG compression, which typically applies heavier quantization to the chrominance components than to the luminance.

2. **Block Division:** The luminance component is divided into non-overlapping  $8 \times 8$  blocks to match the JPEG compression structure.

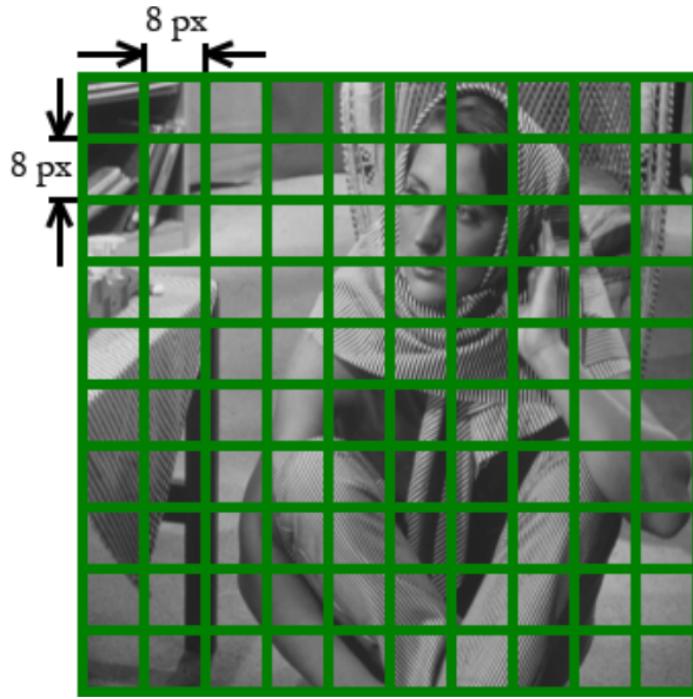


Figure 2.3: DCT deivision image in an  $8 \times 8$  blocks

3. **DCT Application:** Each block is transformed from the spatial domain to the frequency domain using the DCT, producing a matrix of frequency coefficients.

**Frequency Domain Transformation: Block-Based DCT** To achieve robustness against signal processing attacks, the proposed framework employs the Discrete Cosine Transform (DCT). The process follows these steps:

- Division of the image into  $8 \times 8$  pixel blocks.
- Application of the DCT to each block.

**Block Partitioning** The image channel (e.g., the  $Cb$  or  $Cr$  component) is first partitioned into non-overlapping blocks of size  $N \times N$ , where  $N = 8$ . This localization allows the algorithm to handle local image characteristics effectively. Let  $f(i, j)$  represent the intensity value at coordinates  $(i, j)$  within an  $8 \times 8$  block.

**Two-Dimensional DCT (2D-DCT)** For each block, the 2D-DCT is applied to convert the spatial data into the frequency domain. The DCT-II variant is defined as:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cos \left[ \frac{(2i+1)u\pi}{16} \right] \cos \left[ \frac{(2j+1)v\pi}{16} \right] \quad (2.5)$$

where  $u, v$  are the horizontal and vertical frequencies  $u, v \in \{0, 1, \dots, 7\}$ , and the normalization factors  $C(u)$  and  $C(v)$  are defined as:

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \\ 1 & \text{if } k > 0 \end{cases} \quad (2.6)$$

**Coefficient Analysis and Robustness** The resulting  $8 \times 8$  matrix  $F(u, v)$  consists of:

- **DC Coefficient ( $F(0, 0)$ ):** Represents the average intensity of the block. While it holds the most energy, modifying it significantly impacts visual quality.
- **AC Coefficients ( $F(u, v)$  where  $u, v \neq 0$ ):** Represent higher frequency details.

| DC | AC |
|----|----|----|----|----|----|----|----|
| AC |
| AC |
| AC |
| AC |
| AC |
| AC |
| AC |

Figure 2.4: DC and AC coefficient on DCT Transformation

In this research, we target the **mid-frequency coefficients** for data embedding. High-frequency coefficients are often discarded during lossy compression (quantization), while low-frequency/DC coefficients are too sensitive to changes. Mid-frequency embedding provides an optimal balance between imperceptibility and robustness against tampering.

4. **Coefficient Selection:** A predefined set of mid-frequency coefficients is selected for embedding. Low-frequency coefficients are avoided to prevent visible distortion, while high-frequency coefficients are avoided due to their vulnerability to compression.
5. **Data Embedding:** Secret bits are embedded by modifying the selected DCT coefficients using techniques such as coefficient quantization, parity modification, or least significant bit alteration of coefficient values.
6. **Inverse DCT:** After embedding, the modified coefficients are transformed back into the spatial domain using the inverse DCT to reconstruct the stego-image.

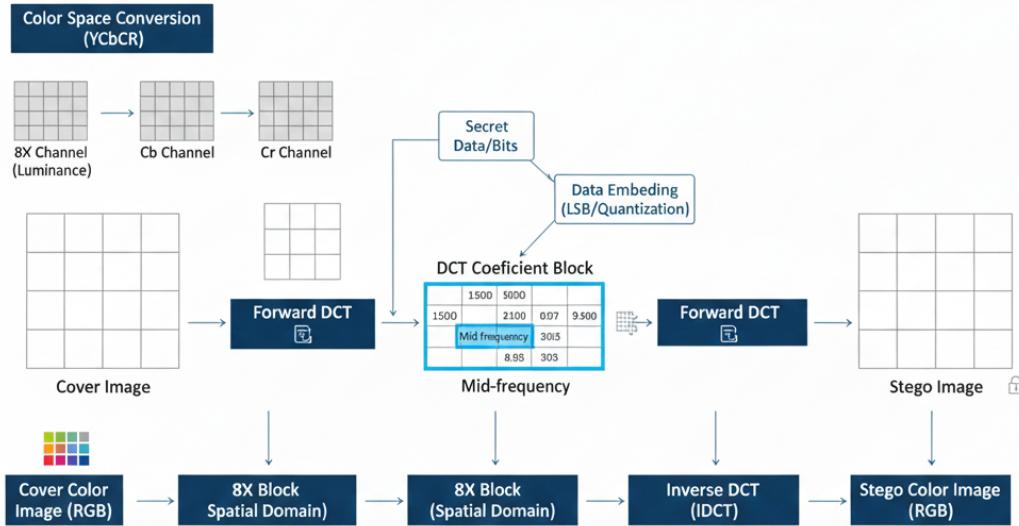


Figure 2.5: Block-based DCT embedding process illustrating transformation, coefficient selection, and inverse transformation.

**Extraction Process** The extraction operation follows the steps in the embedding operation as follows:

1. Conversion of the stego-image to the appropriate color space (from the RGB color space to the YCbCr color space).
2. Application of DCT
3. Identification of the same mid-frequency coefficients used during embedding.
4. Recovery of the hidden bits based on the coefficient modification rule.

Accurate extraction relies on preserving coefficient alignment and consistent block structure between embedding and extraction.

## DCT Method Benefits

compared with the spatial domain method of steganography, DCT offers many benefits:

- Robustness against JPEG compression is significantly improved.
- Embedding information into the frequency domain allows for better imperceptibility than embedding the hidden message into the spatial domain.
- Compatibility with widely used image compression standards.
- DCT-based techniques are less vulnerable to simple noise additions.

## Limitations and Challenges

While DCT-based steganography has significant advantages, many disadvantages exist.

- Have limited robustness when exposed to combined or severe geometric distortions such as cropping, resizing, and rotation.
- Sensitivity to block misalignment caused by image re-sampling.
- The trade-off between payload capacity and robustness is significant.

Although DCT techniques offer substantially increased robustness compared to spatial-domain approaches, DCT alone cannot be relied upon for successful recovery of data under highly adversarial or degradation conditions. [9].

### 2.2.3 Discrete Wavelet Transform (DWT)-Based Steganography

Embedding Data in Images Using DWT-Based Steganography The Discrete Wavelet Transform (DWT) is a popular method for representing images as a set of wavelet coefficients and DWT is used in image steganography because it is a transform-domain technique that allows for the representation of image information at various resolutions. In contrast to the Discrete Cosine Transform, which processes a fixed-size block of an image, the DWT processes an entire image at all frequency bands and all scales simultaneously, thereby providing a means of implementing image steganography with greater perceptual adaptability and resilience to localized distortions than other techniques [10, 11].

#### Overview of the DWT Decomposition

The DWT works by decomposing an image into a set of frequency sub-bands (subbands) by passing the entire image horizontally and vertically through pairs of low-pass and high-pass filters. After a single level DWT decomposition of an image, we end up with four distinct frequency subbands:

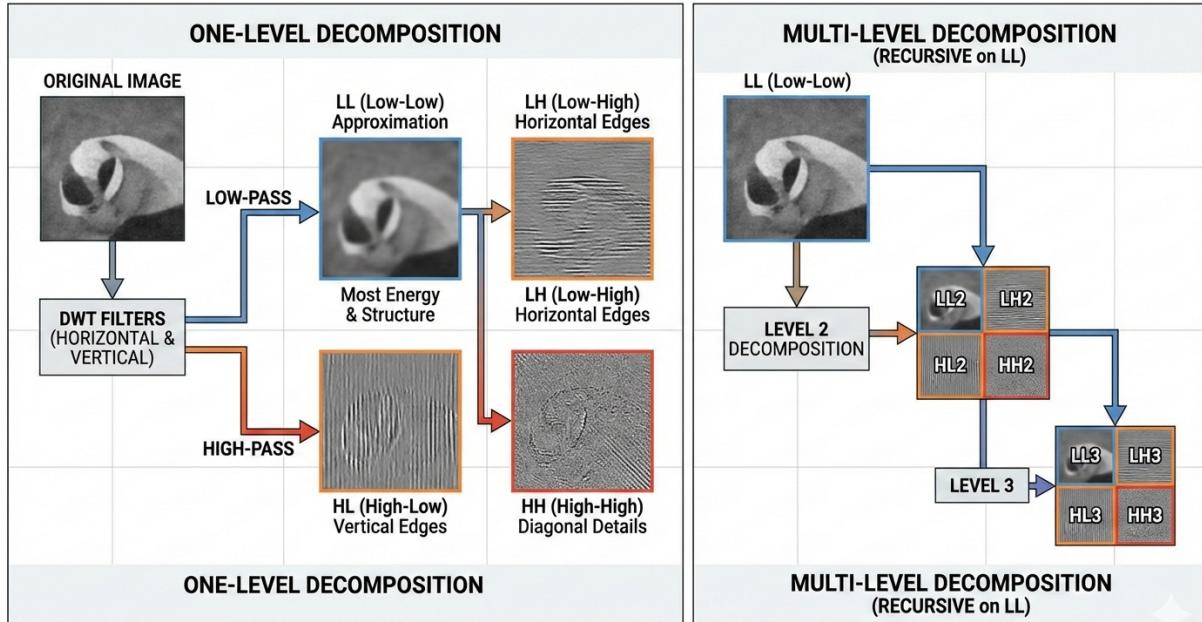
- **LL (Low-Low):** This is the approximation component, which contains the majority of an image's energy and the information regarding its structure.
- **LH (Low-High):** Captures horizontal edge information.
- **HL (High-Low):** Captures vertical edge information.
- **HH (High-High):** Represents diagonal details and fine textures.

The LL sub-band can be decomposed again using the same technique (that is recursion). Consequently, the image-represented data set can be represented in a multi-resolution, or hierarchical, manner.

#### Embedding Process in DWT-Based Steganography

The embedding procedure for DWT-based steganography generally consists of the following steps:

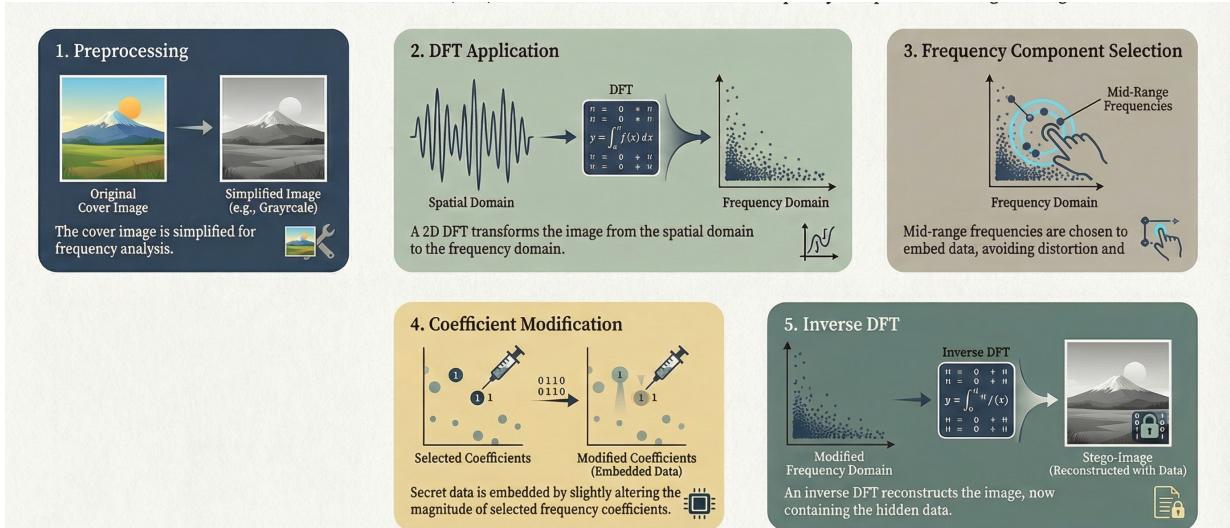
1. **Color Space Selection:** an image is first selected for steganography by choosing between either a color space or a luminance (brightness) component. Color images are generally embedded using the luminance component because it maintains a higher perceptual quality than using one of the color channels.
2. **Wavelet Decomposition:** DWT Decomposition After the luminance has been selected, a discrete wavelet transform (DWT) is applied to multiple levels (i.e., level 1, level 2, etc.) in order to obtain multiple subbands corresponding to each level of the decomposition.



**Figure X:** One-level and multi-level DWT decomposition showing LL, LH, HL, and HH sub-bands.

Figure 2.6: One-level and multi-level DWT decomposition showing LL, LH, HL, and HH sub-bands.

3. **Sub-band Selection:** The subbands for embedding are generally chosen from the higher or mid-level frequency subbands (e.g., LH, HL, HH). The low-frequency subband, LL, may cause strong visual effects on the final output image.
4. **Coefficient Modification:** Modification of Wavelet Coefficients Wavelet coefficients are modified to embed secret data, which can be accomplished in various methods (e.g., coefficient quantization, adjusted threshold, or bit-level).
5. **Inverse DWT:** After modifying the coefficients of the appropriate subbands, an inverted DWT is performed, returning the steganographic image back to the spatial domain.



**Figure 2.7:** DFT-based embedding process illustrating frequency selection, magnitude modification, and inverse transformation

## **Extraction Process**

The process of extracting information through DWT is the reverse of how it was added to the stego-image. The steps for extracting information from the DWT are as follows:

1. To use the same wavelet parameters, perform a DWT on the stego-image
2. Determine which sub-band(s) were used for embedding
3. Use the coefficient modification rule to determine how to retrieve embedded bit(s)
4. Reconstruct the original hidden message from the retrieved bits

The quality of extraction is dependent upon both the wavelet coefficient's stability and also the preservation of the sub-band structure after an image has been degraded.

## **Advantages of DWT-Based Steganography**

DWT-based steganography offers several advantages:

- Multi-Resolution representation is consistent with the way our eyes work.
- Greater Robustness against localized distortions and the cropping of some parts of the image..
- Greater spatial frequency localization than other methods of embedding (block-based).
- The ability to choose where to embed information, depending on the requirements of the application.

## **Limitations and Challenges**

Although DWT Steganography has many benefits, it has a variety of obstacles that may limit its utility. Examples of these limitations include

- The susceptibility of the system to geometric transformations (e.g., resizing/rotating).
- Much higher complexity than methods based on space.
- Potential susceptibility to a multitude of compression attacks or aggressive multi-staged attacks.
- Trade-offs between robustness, imperceptibility, and payload capacity.

These limitations show that DWT techniques provide more robust solutions to Spatial Domain techniques and in some cases may offer advantages over Block-Coding Transform based techniques, however the implementation of other means of ensuring the safety of data recovery may be needed in the event of extreme or compound degradation to the image. [8].

### **2.2.4 Discrete Fourier Transform (DFT)-Based Steganography**

The Discrete Fourier Transform (DFT) represents a fundamental signal processing technique that decomposes an image into its constituent frequency components, providing a global spectral representation of the entire image domain. Unlike block-based transforms such as the Discrete Cosine Transform (DCT), which operate on localized image regions [12], or multi-resolution decomposition techniques like the Discrete Wavelet Transform (DWT) [13], the DFT analyzes the complete image as a unified entity. This holistic frequency representation confers specific advantages for steganographic applications, particularly regarding robustness to geometric transformations including rotation, translation, and scaling operations [14, 15].

The application of DFT in steganography leverages the inherent properties of frequency-domain representations, where imperceptible modifications to specific spectral components can encode information while maintaining perceptual similarity to the original cover medium. This approach has been extensively studied in both theoretical and practical contexts, establishing DFT-based methods as a viable alternative to spatial-domain techniques [16, 17].

## Mathematical Foundation and Fundamental Concepts

**Two-Dimensional DFT Formulation** The two-dimensional DFT transforms a spatial-domain image  $I(x, y)$  of dimensions  $M \times N$  into a complex-valued frequency-domain representation  $F(u, v)$  according to the following definition:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2.7)$$

where  $u$  and  $v$  denote the frequency coordinates in the horizontal and vertical directions, respectively, and  $j = \sqrt{-1}$  represents the imaginary unit. The inverse DFT reconstructs the spatial-domain image through the following transformation:

$$I(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2.8)$$

*Figure Suggestion: Figure X.1:* Mathematical visualization of 2D DFT basis functions showing sinusoidal patterns at various frequencies and orientations.

**Spectral Components and Their Properties** Each complex-valued DFT coefficient  $F(u, v)$  can be decomposed into magnitude and phase components, expressed as:

$$F(u, v) = |F(u, v)| \cdot e^{j\phi(u, v)} \quad (2.9)$$

where  $|F(u, v)|$  represents the magnitude spectrum and  $\phi(u, v)$  denotes the phase spectrum. These components encode distinct perceptual characteristics of the image:

- **Magnitude Spectrum:** Quantifies the strength or energy associated with each frequency component, primarily governing contrast distribution, texture patterns, and overall intensity variations across the image [18].
- **Phase Spectrum:** Encodes the spatial structure, edge locations, and positional relationships between image features. Research has demonstrated that the phase spectrum preserves the majority of perceptual information and structural content [18, 19].

The fundamental observation that phase information dominates human visual perception has profound implications for steganography. Consequently, most DFT-based steganographic schemes modify the magnitude spectrum while carefully preserving phase relationships, thereby maintaining visual fidelity and structural integrity [18, 20].

*Figure Suggestion: Figure X.2:* Comparative visualization demonstrating: (a) original spatial-domain image, (b) magnitude spectrum representation, (c) phase spectrum representation, (d) image reconstructed from magnitude only, (e) image reconstructed from phase only, illustrating the dominance of phase information in perceptual quality.

## Frequency Domain Characteristics and Embedding Regions

**Spatial-Frequency Correspondence** The DFT frequency plane exhibits a systematic organization wherein spatial frequency increases with distance from the origin (DC component). This organization partitions the spectrum into distinct regions with specific perceptual and statistical properties [2]:

- **Low-Frequency Region:** Located near the spectrum center, these components encode smooth intensity variations, background illumination, and coarse image structure. The DC component  $F(0, 0)$  represents the average intensity across the entire image.
- **Mid-Frequency Region:** Situated at intermediate distances from the origin, these components correspond to edges, boundaries, and structural details that define object contours and significant image features.
- **High-Frequency Region:** Positioned at the spectrum periphery, these components capture fine textures, rapid intensity transitions, and high-detail patterns. These regions are particularly susceptible to noise and compression artifacts [9].

*Figure Suggestion:* **Figure X.3:** Annotated DFT magnitude spectrum showing the spatial distribution of low-, mid-, and high-frequency regions, with typical embedding zones highlighted.

**Strategic Region Selection for Data Embedding** The selection of appropriate frequency regions for data embedding constitutes a critical design decision that directly impacts the trade-off between imperceptibility, robustness, and embedding capacity:

- **Low-Frequency Modifications:** Alterations to low-frequency components typically produce highly visible artifacts, including blocking effects, intensity shifts, and perceptual distortions, making this region unsuitable for covert communication [8].
- **High-Frequency Modifications:** While changes in high-frequency regions remain imperceptible, these components are highly vulnerable to common signal processing operations such as low-pass filtering, compression, and additive noise [9].
- **Mid-Frequency Selection:** The mid-frequency region offers an optimal compromise, providing sufficient robustness against moderate signal processing while maintaining imperceptibility. This region has emerged as the preferred embedding domain in most DFT-based steganographic systems [17, 16].

*Figure Suggestion:* **Figure X.4:** Visual comparison of stego-images with data embedded in different frequency regions, demonstrating perceptual quality differences and artifact characteristics.

## Data Embedding Methodology

The embedding procedure in DFT-based steganography follows a systematic multi-stage process designed to minimize perceptual distortion while ensuring reliable data recovery:

**Stage 1: Image Preprocessing** The cover image undergoes preprocessing to optimize it for frequency-domain manipulation:

1. **Color Space Conversion:** For color images, conversion to grayscale or selection of a specific color channel (typically luminance) reduces computational complexity while exploiting reduced human visual sensitivity in certain channels.
2. **Normalization:** Pixel intensity values are normalized to a standard range to ensure consistent transform properties.
3. **Windowing (Optional):** Application of window functions (e.g., Hamming or Hann windows) may reduce spectral leakage effects [21].

**Stage 2: Forward DFT Computation** A two-dimensional DFT is computed using efficient algorithms such as the Fast Fourier Transform (FFT), which reduces computational complexity from  $O(N^4)$  to  $O(N^2 \log N)$  for an  $N \times N$  image [22]. The resulting complex-valued spectrum is typically shifted such that low frequencies are centered for intuitive visualization and processing.

**Stage 3: Embedding Region Selection and Coefficient Identification** A predefined set of frequency coefficients within the mid-frequency region is identified based on one or more of the following criteria:

- Magnitude threshold: coefficients with sufficient energy to withstand modifications
- Frequency band definition: specific radial or angular sectors of the spectrum
- Perceptual significance: regions identified through psychovisual modeling [23]
- Pseudo-random selection: key-based coefficient selection for enhanced security

*Figure Suggestion: Figure X.5:* Illustration of coefficient selection strategies in the DFT domain, showing various masking patterns and selection criteria.

**Stage 4: Magnitude Modification for Data Embedding** Secret data bits are embedded through controlled modifications of the magnitude spectrum using one of several established techniques:

- **Quantization-Based Embedding:** Magnitude values are quantized to discrete levels corresponding to data bits, similar to Quantization Index Modulation (QIM) [24]:

$$|F'(u, v)| = Q \cdot \left\lfloor \frac{|F(u, v)|}{Q} + \frac{b}{2} \right\rfloor \quad (2.10)$$

where  $Q$  is the quantization step,  $b \in \{0, 1\}$  is the bit to embed, and  $\lfloor \cdot \rfloor$  denotes the floor function.

- **Additive Embedding:** A scaled embedding strength parameter modulates the magnitude:

$$|F'(u, v)| = |F(u, v)| + \alpha \cdot b \cdot |F(u, v)| \quad (2.11)$$

where  $\alpha$  controls the embedding strength and typically ranges from 0.01 to 0.1 [25].

- **Multiplicative Embedding:** The magnitude is scaled by a factor dependent on the data bit:

$$|F'(u, v)| = |F(u, v)| \cdot (1 + \alpha \cdot (2b - 1)) \quad (2.12)$$

- **Phase-Preserved Magnitude Adjustment:** Modifications ensure that the phase spectrum remains unchanged to preserve structural information:

$$F'(u, v) = |F'(u, v)| \cdot e^{j\phi(u, v)} \quad (2.13)$$

**Stage 5: Inverse DFT and Stego-Image Generation** The modified frequency spectrum  $F'(u, v)$  undergoes inverse DFT transformation to reconstruct the stego-image in the spatial domain:

$$I'(x, y) = \text{IDFT}\{F'(u, v)\} \quad (2.14)$$

Post-processing steps may include rounding pixel values to integer representations and clipping to valid intensity ranges.

*Figure Suggestion: Figure X.6:* Comprehensive flowchart depicting the complete DFT-based embedding pipeline, from cover image input through preprocessing, DFT computation, magnitude modification, inverse transformation, and stego-image output.

## Data Extraction Methodology

The extraction process operates as the inverse of the embedding procedure and requires precise knowledge of the embedding parameters:

### Extraction Procedure

1. **Stego-Image Reception:** The potentially altered stego-image is received and preprocessed identically to the embedding stage.
2. **Forward DFT Application:** The two-dimensional DFT is computed to obtain the frequency-domain representation.
3. **Coefficient Location:** The same frequency regions and specific coefficients used during embedding are identified using the shared secret key or predetermined pattern.
4. **Data Decoding:** Embedded bits are extracted by analyzing the magnitude values according to the employed embedding method:
  - For quantization-based methods: detecting which quantization level the magnitude occupies
  - For additive/multiplicative methods: comparing magnitudes against thresholds or reference values
5. **Message Reconstruction:** Extracted bits are assembled to reconstruct the original secret message, including any error correction decoding if employed.

**Extraction Reliability Factors** Successful extraction depends critically on several factors:

- **Parameter Consistency:** Identical embedding and extraction parameters (regions, coefficients, quantization steps)
- **Frequency Preservation:** Maintenance of relative relationships among frequency components
- **Synchronization:** Proper alignment and identification of embedding locations
- **Channel Distortion:** Minimal degradation during transmission or storage [8]

*Figure Suggestion: Figure X.7:* Extraction process flowchart showing the symmetric nature of embedding and extraction operations, with emphasis on parameter matching requirements.

### Advantages and Distinctive Properties

DFT-based steganographic techniques offer several compelling advantages that distinguish them from spatial-domain and other transform-domain approaches:

- **Geometric Robustness:** The DFT exhibits inherent invariance properties to certain geometric transformations. Specifically, translation in the spatial domain manifests as phase shifts in the frequency domain without affecting magnitudes, while rotation in the spatial domain produces corresponding rotation in the frequency domain [14, 15]. This property enables embedding schemes resistant to minor geometric distortions.
- **Global Representation:** Unlike block-based methods that may introduce blocking artifacts at tile boundaries, the DFT provides a holistic representation that distributes modifications across the entire image, reducing localized distortions [16].

- **Theoretical Foundation:** The DFT is grounded in well-established Fourier analysis theory, providing rigorous mathematical tools for performance analysis, optimization, and security evaluation [26].
- **Frequency-Selective Embedding:** Precise control over which frequency components are modified enables fine-tuned balancing between imperceptibility and robustness based on application requirements [23].
- **Compatibility with Spread Spectrum:** DFT-based methods integrate naturally with spread spectrum techniques, enhancing security and resistance to detection [17, 17].

These characteristics make DFT-based steganography particularly suitable for applications requiring resilience against geometric transformations, such as print-scan processes or digital-analog-digital conversion chains [27].

### Limitations, Challenges, and Practical Considerations

Despite its advantages, DFT-based steganography faces several significant limitations that constrain its practical applicability:

- **Computational Complexity:** The requirement to process the entire image simultaneously results in substantial computational overhead, particularly for high-resolution images. Even with FFT optimizations, the complexity remains higher than spatial-domain methods [22]. Real-time applications may find this prohibitive.
- **Cropping Vulnerability:** Spatial cropping operations fundamentally alter the frequency representation, as the DFT assumes periodic boundary conditions. Removal of image portions disrupts the global frequency structure, typically destroying embedded information [9].
- **Limited Embedding Capacity:** Compared to spatial-domain LSB techniques or DCT-based approaches that operate on multiple independent blocks, DFT-based methods generally offer lower embedding capacity due to the constraint of modifying only perceptually insignificant frequency components [5].
- **Noise Sensitivity:** Additive noise affects frequency components unpredictably, particularly in high-frequency regions. Careful magnitude modification control is essential to prevent extraction errors under noisy conditions [8].
- **JPEG Compression Vulnerability:** While more robust than pure spatial methods, DFT-based approaches remain susceptible to lossy compression schemes like JPEG, which operate in the DCT domain and may significantly alter DFT coefficients during the compression-decompression cycle [9].
- **Detection Vulnerability:** Statistical analysis of frequency-domain histograms and coefficient distributions may reveal embedding artifacts, particularly when embedding capacity approaches theoretical limits [?].

**Mitigation Strategies** To address these limitations, researchers have proposed various enhancement techniques:

- **Hybrid Approaches:** Combining DFT with other transforms (DWT, DCT) to leverage complementary strengths [11]
- **Error Correction Coding:** Incorporating Reed-Solomon or turbo codes to enhance extraction reliability under distortion [17]

- **Template-Based Synchronization:** Embedding geometric synchronization patterns to recover from limited cropping or scaling [27]
- **Adaptive Embedding:** Adjusting embedding strength based on local image characteristics and frequency content [23]

These observations indicate that while DFT-based steganography offers unique advantages for specific applications, optimal performance in diverse operational scenarios typically requires integration with complementary techniques or strategic application to use cases that align with its strength profile [15, 8].

*Figure Suggestion:* **Figure X.8:** Performance comparison charts showing: (a) imperceptibility metrics (PSNR, SSIM) versus embedding capacity for DFT-based methods compared to spatial and DCT approaches, (b) robustness evaluation against various attacks (JPEG compression, noise, filtering, geometric transformations), demonstrating the trade-offs inherent in DFT-based steganography.

## 2.3 Robustness-Oriented Approaches

Achieving robustness in image steganography requires more than choosing a stable transform domain. It demands a system-level design philosophy that anticipates degradation, structures the payload to survive partial loss, and recovers meaningful information even under adverse conditions. This section presents a comprehensive review of the principal strategies that have been developed toward this goal, including error correction coding, synchronization mechanisms, adaptive embedding, spread-spectrum techniques, fragmentation and redundancy, and the role of deep learning.

### 2.3.1 Error Correction Coding for Steganographic Robustness

A widely adopted strategy for improving robustness against signal-processing attacks is the integration of error correction codes (ECC) into the steganographic payload. Because operations such as JPEG compression, additive noise, and low-pass filtering introduce random bit-level errors into the recovered message, ECC allows the receiver to detect and correct a bounded number of corrupted bits without any knowledge of the original image [28].

**Block codes.** Reed–Solomon (RS) codes are particularly popular in steganographic applications because they operate on multi-bit symbols rather than individual bits, making them well-suited to burst errors introduced by block-based compression [17]. An RS( $n, k$ ) code encodes  $k$  information symbols into  $n$  codeword symbols, enabling the correction of up to  $\lfloor (n - k)/2 \rfloor$  erroneous symbols. BCH codes offer a similar correction capability and have been used where binary codewords are preferable [29].

**Convolutional and turbo codes.** Convolutional codes introduce memory into the encoding process and are decoded efficiently with the Viterbi algorithm, making them suitable for steganographic channels that exhibit correlated noise patterns [?]. Turbo codes, which concatenate two recursive convolutional encoders separated by an interleaver, approach the Shannon capacity limit and have been applied in watermarking systems that demand near-optimal error recovery [17].

**LDPC codes.** Low-density parity-check (LDPC) codes have attracted growing interest in robust steganography because their iterative belief-propagation decoding is highly effective even at low signal-to-noise ratios [?]. Compared with turbo codes, LDPC codes offer lower decoding complexity for long codewords and are therefore better suited to high-resolution image payloads.

**Limitations of ECC alone.** Despite their effectiveness against random bit corruption, ECC schemes share a fundamental assumption: the decoder must receive the codeword bits *in the correct order*. Geometric attacks such as cropping, rotation, and resizing destroy positional correspondence between the embedded and extracted data, producing not random errors but *burst erasures* at unknown locations [28]. Under these conditions, even maximum-distance-separable codes fail because the error positions cannot be identified from the received sequence alone. This observation motivates the use of complementary mechanisms, discussed in the following subsections, that address synchronization rather than bit-level correctness.

### 2.3.2 The Synchronization Problem

Synchronization loss is arguably the most critical and least resolved challenge in robust steganography [30]. A steganographic system is said to be synchronized when the extractor can reliably locate the embedding positions within the received image. Most classical schemes achieve this by relying on fixed coordinate systems, block-partition boundaries, or predetermined pixel sets. Geometric transformations invalidate these references, causing the extractor to read pixels at incorrect positions—a failure mode qualitatively different from ordinary bit corruption.

#### Causes and Manifestations

- **Cropping** removes entire image regions, permanently destroying any fragments embedded within them. Even when the remaining image content is intact, the extractor can no longer determine the original image boundaries and therefore cannot reconstruct the coordinate grid used during embedding [29].
- **Resizing and resampling** change the spatial resolution through interpolation, shifting pixel positions and distorting block boundaries. Because DCT-based schemes depend on exact  $8 \times 8$  block alignment, even a one-pixel shift causes the extractor to operate on misaligned blocks, producing a catastrophic increase in bit error rate [28].
- **Rotation** introduces a global angular offset. Unless the embedding domain is rotation-invariant, the extractor cannot identify coefficient positions and the extraction fails entirely [31].
- **Compound attacks** combine several of the above operations sequentially. Social-media platforms, for example, typically resize images upon upload, apply lossy JPEG compression, and sometimes crop to fixed aspect ratios—all in a single automated pipeline [?]. Each step accumulates synchronization error, making recovery increasingly difficult.

#### Template-Based Synchronization

One practical solution is to embed a known geometric reference pattern—often called a *synchronization template* or *registration mark*—alongside the secret payload. The receiver detects the template, estimates the geometric transformation applied to the image, and inverts it before extraction [30]. Template-based methods are effective against moderate rotations and uniform scaling, but they consume part of the embedding capacity and are vulnerable to cropping if the template region is removed.

Kutter [?] demonstrated that circularly symmetric patterns embedded in the Fourier domain are particularly robust for template-based registration, as they remain detectable after rotation and scaling. Similarly, Solachidis and Pitas [?] proposed embedding ring-shaped patterns in the 2-D DFT magnitude spectrum, exploiting the Fourier shift theorem to achieve translation invariance.

## Feature-Based Alignment

An alternative approach avoids fixed coordinate systems entirely by anchoring embedding positions to stable image features—corners, edges, or salient regions—that can be reliably re-detected after geometric distortion [31]. Feature detectors such as SIFT (Scale-Invariant Feature Transform) have been adopted in watermarking to compute a canonical coordinate frame from the image content itself, enabling extraction without prior knowledge of the applied transformation [29, 28]. The primary limitation is that severe cropping or blurring may destroy the features used for alignment, rendering the approach ineffective under compound attacks.

## Invariant-Domain Embedding

A more fundamental solution embeds data directly in a representation that is algebraically invariant to the class of expected transformations. DFT magnitude coefficients, for instance, are translation-invariant because spatial shifts map to phase changes without affecting magnitudes [18]. Embedding in the log-polar Fourier domain achieves additional invariance to rotation and scaling, a property exploited by several watermarking systems [31, 16]. Radial harmonic transforms and Zernike moments have also been explored as rotation-invariant embedding domains, though their applicability to steganography is limited by their lower embedding capacity compared with block-based transforms [29].

### 2.3.3 Spread-Spectrum Steganography

Spread-spectrum (SS) techniques originate from secure communications and were adapted for image steganography by Marvel et al. [17], whose work remains a foundational reference in robust data hiding. The core idea is to distribute each secret bit across a large number of cover elements (pixels or transform coefficients) using a pseudo-random spreading sequence, so that the energy of the hidden signal is spread too thinly to be concentrated by any localized attack.

Formally, let  $\mathbf{m} \in \{-1, +1\}^K$  be the message after bipolar encoding, and let  $\mathbf{w}_k \in R^N$  be a unit-norm pseudo-random spreading vector for the  $k$ -th bit, drawn from a secret key. The stego-signal added to the cover image  $\mathbf{c}$  is:

$$\mathbf{s} = \mathbf{c} + \alpha \sum_{k=1}^K m_k \mathbf{w}_k, \quad (2.15)$$

where  $\alpha > 0$  is an embedding strength parameter that trades imperceptibility against robustness [17, 28]. Extraction correlates the received signal against each spreading vector:

$$\hat{m}_k = \text{sign}(\mathbf{r}^\top \mathbf{w}_k), \quad (2.16)$$

where  $\mathbf{r}$  is the received (possibly degraded) signal. Because the spreading vectors are orthogonal—or nearly so for long sequences—inter-symbol interference is minimal, and the correlation effectively averages out additive noise.

**Informed embedding and dirty-paper coding.** A significant theoretical advance over basic SS steganography came with the introduction of *informed embedding*, in which the embedding process exploits knowledge of the cover signal to minimize distortion for a given robustness level [24]. Quantization Index Modulation (QIM), proposed by Chen and Wornell [24], selects a quantization lattice based on the secret bit and quantizes the coefficient to the nearest point on that lattice. QIM achieves the *writing on dirty paper* capacity, meaning it can approach the theoretical maximum robustness for a given distortion budget, and it is now widely used in DCT-domain watermarking [25, 3].

**Limitations of spread-spectrum methods.** Although SS techniques are highly robust against additive noise and moderate compression, they share the synchronization vulnerability of other location-dependent schemes: geometric attacks that shift the spreading vector grid cause the correlation in Equation (2.16) to collapse [28]. Moreover, the per-bit capacity of SS steganography is low, as each bit requires  $N \gg 1$  coefficients for reliable spreading, limiting the practical payload size.

### 2.3.4 Adaptive Embedding and Perceptual Masking

Adaptive embedding methods adjust the embedding strength locally based on image content, exploiting perceptual masking properties of the human visual system (HVS) to embed more information where distortion is less visible and less where it is more noticeable [23]. This strategy simultaneously improves imperceptibility and, in some implementations, robustness, because stronger embedding in textured or edge-rich regions is harder for processing operations to fully suppress.

#### Texture and Edge Masking

The HVS is less sensitive to luminance changes in regions of high spatial activity—such as textured surfaces, edges, and complex patterns—than in flat, homogeneous areas [2]. Adaptive steganographic methods leverage this by computing a local activity measure  $\sigma^2(b)$  for each image block  $b$  (e.g., the variance of DCT AC coefficients) and scaling the embedding strength accordingly:

$$\alpha(b) = \alpha_0 \cdot f(\sigma^2(b)), \quad (2.17)$$

where  $f(\cdot)$  is a monotonically increasing function and  $\alpha_0$  is a base strength parameter [23, 28]. Blocks with higher texture activity receive stronger embedding, increasing local robustness in those regions while limiting visible distortion in smooth areas.

#### Frequency-Selective Masking

Complementary to spatial masking is frequency-domain masking, in which the embedding energy is concentrated in mid-frequency DCT or DWT coefficients that fall within the HVS sensitivity range [25]. The JPEG quantization matrix provides a useful proxy for perceptual importance: coefficients assigned a larger quantization step are less perceptually significant and can therefore tolerate stronger embedding [29]. By aligning embedding choices with the JPEG quantization structure, adaptive DCT methods also gain a secondary benefit: the embedded signal is less likely to be removed by JPEG compression, because only the coefficients that survive compression at the target quality factor are used for embedding.

#### Just-Noticeable Difference Models

More principled approaches model the maximum imperceptible modification at each coefficient using just-noticeable difference (JND) thresholds derived from psychophysical experiments [28]. A JND model  $J(u, v)$  specifies the maximum change to coefficient  $F(u, v)$  that remains below human detection threshold. Embedding is then constrained so that  $|\Delta F(u, v)| \leq J(u, v)$ , guaranteeing imperceptibility by construction. Podilchuk and Zeng [?] demonstrated that JND-constrained watermarking achieves significantly better imperceptibility-robustness trade-offs than fixed-strength embedding, and their framework has been adapted for steganographic applications.

### 2.3.5 Fragmentation, Redundancy, and Distributed Embedding

A conceptually distinct approach to robustness abandons the goal of protecting individual embedding positions and instead focuses on *payload survivability* through structured redundancy and spatial distribution. Rather than assuming that the entire image is preserved, these methods acknowledge

that portions of the cover may be destroyed and design the embedding strategy to permit recovery from whatever fragments remain [29, 28].

## Redundant Embedding

The simplest instantiation of this idea is to embed the same payload multiple times across non-overlapping image regions [7]. If the image is divided into  $R$  non-overlapping regions and each carries a full copy of the payload, then recovery is possible as long as at least one region survives intact. While effective against cropping of a single contiguous area, naive redundant embedding does not scale well with payload size, as the effective capacity is reduced by a factor of  $R$ .

## Erasure-Correcting Codes and Fountain Codes

A more capacity-efficient approach uses *erasure codes*, which are designed for channels that lose entire blocks rather than individual bits. Reed–Solomon codes can be used in erasure mode: an RS( $n, k$ ) code can recover the  $k$  source symbols from *any*  $k$  of the  $n$  transmitted symbols, regardless of which  $n - k$  are erased [3]. Fountain codes—including LT codes and Raptor codes—go further by generating a potentially unlimited number of encoded symbols such that the original message can be recovered from any sufficiently large subset [3]. Applying fountain codes to image steganography means that recovery is possible from any portion of the stego-image that survives cropping or localized tampering, provided the surviving region contains enough encoded fragments.

## Inter-Fragment Dependency Graphs

Beyond simple redundancy, structured payload representations can encode cross-fragment relationships so that surviving fragments carry partial information about missing ones. This transforms the reconstruction problem into a belief-propagation or message-passing problem on a factor graph, where each received fragment provides constraints on the values of neighboring fragments [28]. The design of the dependency graph—its degree distribution, connectivity, and redundancy level—directly controls the trade-off between capacity overhead and recovery probability under various patterns of fragment loss.

## Multi-Domain and Multi-Channel Distribution

Embedding fragments across different transform domains provides complementary robustness: spatial-domain fragments survive geometric distortions better than frequency-domain fragments, while DCT-domain fragments are more robust to compression than pixel-level modifications [32, 29]. Similarly, distributing redundant copies across the luminance and chrominance channels of a color image exploits the fact that different channels are affected differently by common processing operations [25]. A combined multi-domain, multi-channel strategy therefore reduces the probability that all copies of any given fragment are simultaneously destroyed.

### 2.3.6 Robust Steganography vs. Digital Watermarking

Robust steganography and digital watermarking share a common toolbox—transform-domain embedding, ECC, spread spectrum, adaptive strength—but differ fundamentally in their objectives, threat models, and success criteria [28, 33].

Digital watermarking is designed for *ownership verification* and *content authentication*: the embedded mark is expected to be detectable by an authorized verifier, and its persistence under attack is the primary success criterion. The presence of a watermark is not concealed; on the contrary, watermarking systems often include public detection algorithms that any party can use to verify authenticity [29]. Consequently, watermarking schemes can afford to use stronger, more visible embedding, and they are evaluated primarily on detectability under attack (measured by Normalized Correlation or bit detection rate) rather than on undetectability.

Steganography, by contrast, requires *covert communication*: the existence of the hidden message must remain undetectable to unauthorized observers, even sophisticated steganalysts [34, 7]. This imposes a fundamentally different constraint on the embedding process. Techniques that enhance robustness—redundancy, high embedding strength, structured payload distribution—tend to introduce statistical artifacts that increase the risk of steganalytic detection [5]. The steganographer must therefore balance three competing objectives simultaneously: imperceptibility, robustness, and undetectability (resistance to steganalysis), whereas a watermarker need only balance the first two.

Robust steganography occupies the intersection of these paradigms, inheriting the covertness requirement of classical steganography and the survivability requirement of watermarking. This dual constraint is what makes robust steganographic design particularly challenging and motivates the payload-oriented framework proposed in Chapter 3.

### 2.3.7 Steganalysis and the Detectability–Robustness Trade-off

A complete treatment of robustness must acknowledge the threat of steganalysis—the statistical detection of hidden communication—because robustness-enhancing design choices often increase detectability [5, 29].

#### Classical Steganalysis

Early steganalysis methods focused on statistical anomalies introduced by specific embedding algorithms. Chi-squared analysis [34] detects the pairing of adjacent pixel values caused by LSB substitution. RS analysis examines the ratio of regular and singular pixel groups, which is disrupted predictably by LSB embedding [34]. These attacks are *specific*: they exploit known properties of a single embedding algorithm and fail against others.

#### Universal Feature-Based Steganalysis

Modern steganalysis is *universal*: it extracts a high-dimensional feature vector from the image and trains a machine learning classifier to distinguish cover images from stego-images, without requiring knowledge of the specific embedding algorithm [5]. The rich model of Fridrich and Kodovský [5] computes co-occurrence statistics of pixel prediction errors in multiple directions, producing feature vectors with tens of thousands of dimensions that capture subtle embedding artifacts across a wide range of steganographic methods. Paired with ensemble classifiers, these rich feature sets achieve near-perfect detection even at very low embedding rates, highlighting the difficulty of simultaneously achieving robustness and undetectability.

#### Implications for Robust Steganographic Design

The detectability–robustness trade-off imposes a practical ceiling on the embedding strength that can be used in a steganographic (as opposed to watermarking) context. Stronger embedding improves survivability under attack but leaves larger statistical footprints in the stego-image. Several studies have shown that the embedding strength required for robustness against JPEG compression at quality factor 70 already produces detectable artifacts in standard steganalytic feature spaces [?, 29]. This motivates design approaches that achieve robustness through *structure*—fragmentation, cross-fragment relationships, and domain diversity—rather than through raw embedding strength, as structural approaches can maintain a lower per-coefficient distortion while still enabling recovery under partial data loss.

### 2.3.8 Deep Learning–Based Robust Steganography

Recent years have seen the application of deep learning to steganographic embedding and extraction, with the goal of jointly optimizing all three objectives—imperceptibility, capacity, and robust-

ness—through end-to-end training on large image datasets [35, 36].

### Encoder–Decoder Architectures

The dominant paradigm is an encoder–decoder architecture in which a convolutional neural network (CNN) encoder embeds the payload into the cover image, and a paired CNN decoder recovers it [35]. The HiDDeN framework of Zhu et al. [35] introduces a differentiable noise layer between the encoder and decoder during training, simulating common attacks (JPEG compression, Gaussian noise, cropping) so that the network learns embeddings that survive them. The adversarial component of HiDDeN adds a discriminator network that penalizes statistically detectable stego-images, incorporating a steganalysis-like signal directly into the training objective.

### Generative Adversarial Approaches

Hayes and Danezis [36] proposed a generative adversarial framework in which a steganographer network and a steganalyst network compete, with the steganographer learning to produce stego-images that are simultaneously imperceptible, robust, and undetectable. This adversarial training paradigm has since been extended by multiple groups, with improvements in robustness against geometric attacks through the use of spatial transformer networks and learned re-synchronization modules [37].

### Limitations in the Context of This Work

Despite promising results under trained conditions, deep learning–based approaches present several limitations that reduce their suitability for the robustness objectives of this thesis.

1. **Distribution shift.** Neural steganographic systems are robust only to the specific attacks simulated during training. Exposure to unseen attack combinations or parameter values outside the training distribution causes unpredictable and often catastrophic performance degradation [37].
2. **Opacity.** Unlike classical transform-domain methods, where embedding positions, redundancy levels, and reconstruction strategies are explicitly defined and analyzable, neural networks operate as black-box models. This opacity makes it impossible to reason analytically about synchronization behavior, payload survivability under geometric attacks, or the probability of partial recovery under a given loss model.
3. **Resource requirements.** Training competitive deep steganographic models requires large-scale datasets and substantial computational infrastructure, limiting applicability in constrained or low-resource environments.
4. **Interpretability of failure.** When a deep steganographic system fails to recover the payload, there is no principled mechanism to determine how much of the payload survived or to reconstruct a partial message. The binary succeed/fail nature of neural decoding is incompatible with the partial-recovery objective of this thesis.

For these reasons, this thesis adopts a classical, model-driven framework rather than a data-driven one, while acknowledging deep learning steganography as a promising and active research direction.

#### 2.3.9 Discussion and Research Gap

The techniques reviewed in this section reveal a consistent pattern: existing robustness strategies address either *bit-level corruption* (through ECC and spread spectrum) or *limited geometric distortion* (through synchronization templates and invariant domains), but few systems are designed to

tolerate the *compound* of geometric desynchronization, partial data erasure, and signal-level noise that characterizes real-world uncontrolled environments.

Table 2.1 summarizes the strengths and limitations of each major category of robustness-oriented approach.

Table 2.1: Comparison of robustness-oriented steganographic approaches.

Approach	Signal attacks	Geometric attacks	Partial recovery	Capacity
LSB (spatial)	Poor	Poor	No	High
DCT mid-freq.	Moderate	Poor	No	Medium
DWT sub-band	Moderate	Moderate	No	Medium
DFT magnitude	Moderate	Good	No	Low
ECC alone	Good	Poor	No	Reduced
Spread spectrum	Good	Poor	No	Low
Template sync.	Moderate	Moderate	No	Reduced
Redundant embedding	Moderate	Moderate	Partial	Low
Fragmentation + erasure	Good	Good	Yes	Configurable
Deep learning	Good	Variable	No	Medium

The critical observation from Table 2.1 is that no existing single technique achieves both good robustness against compound attacks *and* the ability to partially reconstruct the hidden message when some data is lost. This gap—between robustness as the survival of embedding locations and robustness as the survival of *meaning*—motivates the self-recovering framework proposed in Chapter 3. By combining structured payload fragmentation, multi-domain distribution, erasure-code redundancy, and damage-aware extraction, the proposed system is designed to maximize the probability of recovering usable information from whatever portion of the stego-image survives, without requiring perfect synchronization or complete data preservation.

## 2.4 Attacks and Image Degradation Models

In real-world communication environments, stego-images are rarely transmitted or stored in their original form. Instead, they are frequently subjected to various distortions caused by image processing operations, platform-specific optimizations, or intentional tampering. These distortions can be modeled as attacks that aim, either deliberately or unintentionally, to disrupt the embedded information. A robust steganographic system must therefore be evaluated under realistic attack models that reflect practical usage scenarios rather than ideal transmission conditions [7, 8].

From a robustness perspective, image attacks can be broadly classified into signal processing attacks and geometric attacks. While both categories threaten the integrity of hidden data, they affect steganographic systems in fundamentally different ways.

### 2.4.1 Signal Processing Attacks

Signal processing attacks modify the pixel intensity values of an image while preserving its overall geometric structure. These attacks primarily affect the frequency content of the image and pose a serious challenge to steganographic methods that rely on fragile pixel-level or coefficient-level modifications [6].

One of the most common signal processing attacks is lossy compression, particularly JPEG compression. JPEG operates by transforming the image into the DCT domain and applying quantization

to the frequency coefficients. During this process, high-frequency and some mid-frequency components are significantly altered or discarded, which can lead to partial or complete loss of embedded data, especially when the hidden information is stored in unstable coefficients [12, 5]. Even moderate compression levels can introduce substantial bit errors in conventional steganographic schemes.

Additive noise represents another significant class of signal processing attacks. Noise sources such as Gaussian noise or salt-and-pepper noise introduce random perturbations to pixel values, which propagate into the transform domain and distort the embedded signal. These random variations often result in bit-level corruption, reducing extraction accuracy and increasing the bit error rate [8]. Noise attacks are particularly harmful to low-strength embedding schemes designed primarily for imperceptibility.

Filtering operations, including low-pass, median, and smoothing filters, further degrade embedded information by attenuating high-frequency components and suppressing fine image details. Since many steganographic techniques embed data in frequency bands associated with edges or textures, filtering can significantly weaken or erase the hidden signal [2]. Collectively, signal processing attacks introduce cumulative distortions that can severely compromise data recovery when multiple operations are applied sequentially.

#### 2.4.2 Geometric Attacks

Geometric attacks alter the spatial structure of an image, resulting in changes to pixel positions and spatial relationships. Unlike signal processing attacks, geometric attacks do not necessarily destroy the embedded data itself but instead disrupt the synchronization between the embedding and extraction processes [32].

Cropping is one of the most destructive geometric attacks in the context of robust steganography. By removing portions of the image, cropping leads to irreversible loss of embedded fragments located within the removed regions. For block-based or location-dependent embedding schemes, cropping causes partial data loss and prevents correct reconstruction of the hidden message [7].

Resizing and scaling operations modify the spatial resolution of the image through interpolation or resampling. These operations change pixel alignment and block boundaries, making it difficult to locate the original embedding positions during extraction. Even when the embedded data remains partially intact, resampling introduces interpolation artifacts that degrade extraction reliability [15].

Rotation introduces additional complexity by altering the orientation of spatial features. Rotational transformations cause global misalignment, which is particularly problematic for steganographic methods that rely on fixed coordinate systems or block-based partitioning. Without explicit synchronization mechanisms or invariant representations, rotation typically renders extraction infeasible [8].

### Steganalysis and Detectability Considerations

In addition to robustness against image degradation, steganographic systems are also subject to intentional detection attacks, commonly referred to as steganalysis. Steganalysis aims to determine whether a given image contains hidden information by exploiting statistical inconsistencies, structural artifacts, or deviations from natural image models introduced during the embedding process [6, 5]. Modern steganalysis techniques employ a wide range of methods, including histogram analysis, feature-based classifiers, and machine learning approaches, which can effectively detect many high-capacity or poorly designed steganographic schemes.

It is important to distinguish between robustness and detectability in the context of steganographic design. Robust steganography focuses on preserving the embedded payload under image degradation and tampering, whereas steganalysis resistance aims to minimize the statistical detectability of hidden communication. These objectives are often conflicting, as strategies that improve robustness—such as redundancy, stronger embedding, or structured payload distribution—may increase the risk of detection by altering statistical properties of the cover image [8].

This thesis does not aim to address steganalysis resistance explicitly. Instead, the primary focus is on enhancing payload survivability under realistic signal processing and geometric distortions. While detectability is an important aspect of steganographic security, it is treated as a secondary concern in this work. The inherent trade-off between robustness and undetectability is acknowledged, and the proposed framework prioritizes reliable data recovery in hostile environments over minimizing statistical traces of embedding.

### 2.4.3 Impact of Attacks on Steganographic Robustness

From a system-level perspective, signal processing attacks primarily result in bit-level corruption of embedded data, whereas geometric attacks cause desynchronization between the embedding and extraction stages. While error correction coding and redundancy can partially mitigate the effects of signal processing distortions, synchronization loss caused by geometric attacks remains a major unresolved challenge in conventional steganographic systems [32, 5].

Furthermore, real-world scenarios often involve compound attacks, such as compression followed by cropping or resizing, which exacerbate the weaknesses of traditional embedding strategies. These observations highlight the limitations of approaches that focus solely on protecting embedding locations and motivate the need for robustness-oriented frameworks that emphasize payload survivability and partial recovery under severe and combined degradation conditions.

## 2.5 Discussion and Research Gap

The literature reviewed in this chapter demonstrates significant progress in the field of image steganography, particularly in the development of transform-domain techniques and robustness-oriented embedding strategies. Methods based on DCT, DWT, and DFT have shown improved resistance to common signal processing operations compared to spatial-domain approaches, especially under mild compression and noise conditions [8]. However, despite these advancements, existing techniques continue to exhibit fundamental limitations when deployed in realistic and uncontrolled digital environments.

A primary limitation of conventional steganographic systems lies in their vulnerability to compound attacks. While many methods are evaluated against individual distortions such as JPEG compression or additive noise, real-world scenarios frequently involve sequential or combined operations, including compression followed by resizing or cropping. Under such conditions, robustness rapidly degrades, revealing the inadequacy of approaches that focus on isolated attack resistance rather than holistic survivability.

Robustness-oriented techniques, including the use of error correction codes and controlled redundancy, have been widely adopted to mitigate bit-level corruption caused by signal processing attacks [8]. Although these methods can significantly reduce extraction errors under noise and compression, they fail to address the synchronization problem introduced by geometric transformations. Cropping, resizing, and rotation disrupt spatial alignment and block correspondence, rendering the embedded data inaccessible even when it remains partially present within the image [32]. This desynchronization represents a fundamental challenge that cannot be resolved through redundancy alone.

Several synchronization-aware solutions have been proposed, such as invariant-domain embedding, feature-based alignment, and template-based synchronization markers [31]. While these approaches improve resistance to limited geometric distortions, they often introduce substantial overhead, reduce embedding capacity, or remain vulnerable to severe or compound transformations. Consequently, existing systems remain heavily dependent on accurate localization of embedding positions, making them fragile in the presence of uncontrolled spatial modifications.

Another critical observation from the literature is the conceptual overlap between robust steganography and digital watermarking. Although both domains employ similar robustness-enhancing techniques, their objectives differ fundamentally. Watermarking prioritizes persistent detectability for

ownership verification, whereas steganography requires covert communication with minimal statistical detectability [8, 6]. Many robustness-driven methods implicitly shift toward watermarking paradigms, weakening the steganographic requirement of secrecy while still failing to guarantee data recovery under severe distortions.

Furthermore, intentional detection attacks in the form of steganalysis present an additional challenge. Modern steganalysis techniques exploit statistical anomalies and learned feature representations to identify hidden communication [5]. The literature indicates a clear trade-off between robustness and undetectability, as strategies that enhance survivability—such as stronger embedding or redundancy—may increase statistical detectability. Although steganalysis resistance is not the primary focus of this thesis, its existence highlights the need for carefully balanced design choices in robust steganographic systems.

Recent deep learning-based steganographic approaches attempt to address some of these challenges through data-driven optimization [36, 37]. While these methods demonstrate promising performance under trained conditions, their reliance on large datasets, lack of interpretability, and sensitivity to unseen or compound attacks limit their applicability for controlled robustness analysis. Moreover, learned models provide limited insight into synchronization behavior and payload survivability under geometric distortions, which remain central challenges in robust steganography.

Collectively, these observations reveal a critical research gap in existing work. Current steganographic systems predominantly focus on protecting embedding locations and coefficients rather than ensuring the survivability of the hidden payload itself. Robustness is often evaluated in terms of local signal preservation, while the ability to recover meaningful information under partial data loss and desynchronization remains largely unexplored.

Figure 2.X: Comparison between embedding-centric robustness and payload-oriented survivability

Figure 2.Y: Impact of signal processing and geometric attacks on synchronization and data recovery

This thesis addresses this gap by shifting the design focus from embedding-centric robustness to payload-oriented survivability. Instead of assuming perfect synchronization or complete data preservation, the proposed framework emphasizes structured message fragmentation, controlled redundancy, and multi-domain distribution to enable partial recovery under severe and compound image degradation. By tolerating data loss and synchronization imperfections, the framework aims to achieve reliable covert communication in hostile and uncontrolled digital environments.

## 2.6 Recent Advances in Deep Learning-Based Steganography

The preceding sections have reviewed the classical foundations of robust image steganography, covering spatial-domain techniques, frequency-domain transforms (DCT, DWT, DFT), and robustness-oriented design strategies developed primarily between the late 1990s and the mid-2010s. Since 2018, however, the field has undergone a dramatic transformation driven by the application of deep learning. Convolutional neural networks (CNNs), generative adversarial networks (GANs), and attention-based architectures have been adapted to the steganographic task, promising simultaneous optimisation of imperceptibility, payload capacity, and robustness in ways that classical hand-crafted methods cannot easily achieve.

This section surveys the most influential deep learning-based steganographic systems published between 2018 and 2024, with particular attention to their robustness properties and the extent to which they address the payload-survivability problem that motivates the present work. The section concludes with a critical assessment of the limitations of data-driven approaches and an explicit justification for the classical model-driven framework adopted in this thesis.

### 2.6.1 The Shift Toward End-to-End Learned Steganography

Classical steganographic systems are hand-engineered: the designer chooses a transform domain, selects embedding positions, fixes an embedding rule, and derives extraction analytically from the same

rule. Every component is transparent and interpretable. The limitation is that design decisions are made independently and are therefore individually sub-optimal; in particular, embedding positions and strengths are chosen without direct knowledge of how a specific attack will affect the received signal.

Deep learning approaches to steganography replace this modular pipeline with a single differentiable system trained end-to-end on a large corpus of natural images [35, 38]. In the typical architecture, an *encoder* network  $E_\theta$  maps a cover image  $C$  and a secret message  $M$  to a stego-image  $S$ :

$$S = E_\theta(C, M), \quad (2.18)$$

while a *decoder* network  $D_\phi$  recovers the message from a (possibly degraded) received signal  $\tilde{S}$ :

$$\hat{M} = D_\phi(\tilde{S}). \quad (2.19)$$

Training minimises a composite loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{img}}(C, S) + \lambda_2 \mathcal{L}_{\text{msg}}(M, \hat{M}) + \lambda_3 \mathcal{L}_{\text{adv}}, \quad (2.20)$$

where  $\mathcal{L}_{\text{img}}$  penalises perceptible distortion between the cover and stego-images,  $\mathcal{L}_{\text{msg}}$  penalises errors in the recovered message, and  $\mathcal{L}_{\text{adv}}$  is an optional adversarial term from a discriminator trained to detect the stego-image [36]. The weights  $\lambda_1, \lambda_2, \lambda_3$  control the balance between the three steganographic objectives. The key advantage over classical methods is that gradient back-propagation allows the encoder and decoder to co-adapt: the encoder learns to embed information in regions that the decoder can reliably read, even after passing through a differentiable noise model that simulates real-world attacks during training.

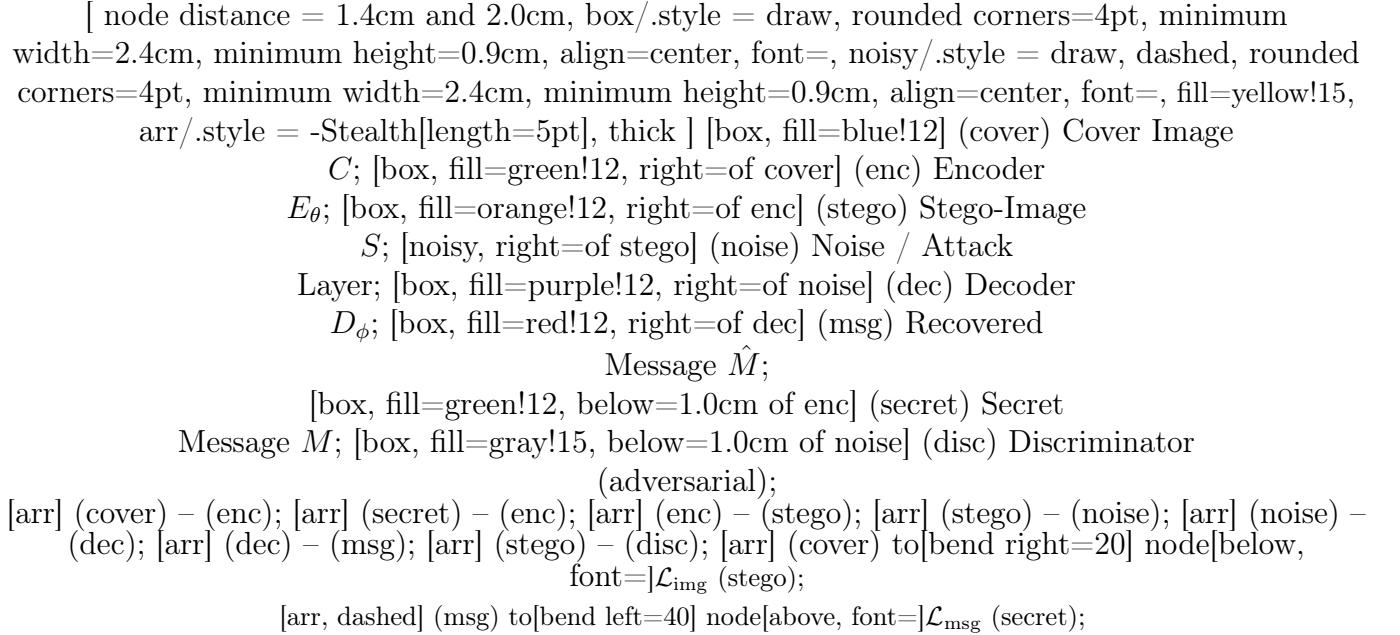


Figure 2.8: Generic encoder–decoder architecture for deep steganography. The noise layer simulates signal processing and geometric attacks during training, forcing the encoder to learn embeddings that survive realistic degradation. An optional discriminator provides an adversarial loss  $\mathcal{L}_{\text{adv}}$  to penalise statistically detectable stego-images.

## 2.6.2 HiDDeN: The Foundational Framework (2018)

The most influential early work in the deep steganography paradigm is HiDDeN (*Hiding Data with Deep Networks*) by Zhu et al. [35], presented at ECCV 2018. HiDDeN introduced three key ideas that have since become standard in the field.

**Architecture.** HiDDeN employs fully convolutional encoder and decoder networks. The encoder is a 7-layer CNN that takes the cover image and a binary secret message (replicated to match the spatial dimensions of the image) and outputs a stego-image of the same resolution. The decoder is an independent CNN that maps a received stego-image directly back to the binary message without any positional information. Neither network uses explicit transform-domain computation; the optimal embedding strategy is learned implicitly from data.

**Differentiable noise layer.** The central contribution of HiDDeN is the insertion of a differentiable noise layer between the encoder and decoder during training. The noise layer is a parametric module that randomly applies one or more of the following operations to the stego-image before it reaches the decoder: JPEG compression (approximated by a differentiable relaxation), Gaussian noise addition, dropout (random pixel zeroing, simulating cropping of small regions), and Gaussian blurring. Because all operations are differentiable with respect to their inputs, gradients flow back through the noise layer to the encoder during training, causing the encoder to learn embeddings that are specifically resistant to the attacks it will encounter at inference time. This is a fundamental architectural insight that distinguishes HiDDeN from classical ECC-based approaches: rather than correcting errors after they occur, the system learns to avoid them at the point of embedding.

**Adversarial imperceptibility.** HiDDeN adds a discriminator network trained to classify images as cover or stego. The discriminator loss is added to the training objective of the encoder, creating a min-max game that encourages the encoder to produce stego-images that are statistically indistinguishable from natural photographs. This is conceptually similar to the GAN training paradigm introduced by Goodfellow et al. [39], applied to the steganographic context.

**Reported performance.** On the COCO dataset, HiDDeN achieves a PSNR above 33 dB for 30-bit payloads under JPEG compression and Gaussian noise, with bit error rates below 1%, significantly outperforming the classical F5 algorithm [40] under equivalent attack conditions.

The limitations of HiDDeN, however, are equally significant. The noise layer must be fixed before training; if the deployed system encounters attacks not represented in the noise layer (e.g. spatial warping, format conversion chains, or aggressive cropping of large image fractions), the encoder has learned no strategy to resist them. This is the distribution-shift problem that recurs throughout deep steganography.

### 2.6.3 SteganoGAN: High-Capacity GAN-Based Steganography (2019)

Zhang et al. [41] introduced SteganoGAN in 2019, extending the HiDDeN framework with a focus on maximising embedding capacity while preserving perceptual quality. SteganoGAN proposes three encoder architectures of increasing depth — *basic*, *residual*, and *dense* — corresponding to different trade-offs between capacity and visual quality. The dense encoder, inspired by DenseNet [42], concatenates feature maps from all preceding layers, allowing the network to leverage both low-level texture features and high-level semantic features when deciding where and how strongly to embed each payload bit.

**Key contributions.** The primary innovation in SteganoGAN is a rigorous treatment of the capacity–quality trade-off. The authors report that their dense encoder achieves capacities of up to 4.4 bpp (bits per pixel) at a PSNR of 30.5 dB, compared to approximately 1.5 bpp for HiDDeN at equivalent quality. The critical observation for the present work, however, is that SteganoGAN *does not include a noise layer*: training is performed under ideal (no-attack) conditions. The resulting stego-images are therefore visually excellent but highly fragile. Under JPEG compression at quality factor 70, the bit error rate of a SteganoGAN-trained decoder rises above 40%, which constitutes complete communication failure. This confirms that capacity-oriented deep models without explicit robustness training are unsuitable for uncontrolled transmission channels.

**Relevance to this thesis.** SteganoGAN illustrates a general principle that pervades both classical and deep steganography: in the absence of specific robustness mechanisms, higher capacity invariably leads to lower survivability. The dense architectures of SteganoGAN produce visually imperceptible embeddings but exploit fragile, high-frequency image regions for extra capacity, exactly the regions that are discarded by lossy compression.

### 2.6.4 ReDMark: Robustness-First Deep Steganography (2021)

Recognising the limitations of capacity-first approaches, Das et al. [43] introduced ReDMark (*Re-distributing the Weights for Robustness Deep Steganographic Marking*) in 2021, the first deep steganographic system explicitly designed around the robustness objective as the primary design constraint.

**Architecture and training.** ReDMark employs a U-Net-style encoder [44] with skip connections that propagate spatial information from the cover image directly to the deeper layers of the embedding network. This architectural choice is motivated by the observation that skip connections encourage the encoder to modulate existing image structures rather than superimpose an independent embedding signal, resulting in stego-images that survive compression better because the embedding is “anchored” to stable image features that compression itself preserves.

The training pipeline of ReDMark incorporates a differentiable JPEG approximation module [45] that accurately models the quantisation artifacts produced by the standard JPEG codec. Additionally, a spatial transformer network (STN) [46] is inserted in the noise layer to simulate mild geometric transformations (rotation by up to 5 and scaling by up to 20%), allowing the decoder to learn a degree of geometric invariance.

**Loss function and weight redistribution.** The distinctive contribution of ReDMark is a dynamic loss-weight redistribution scheme. During training, the weights  $\lambda_1$  and  $\lambda_2$  in Equation (2.20) are adjusted adaptively based on the current attack intensity estimated from a running average of recent BER values. When the bit error rate under the noise layer is high, the training procedure temporarily increases  $\lambda_2$  (message fidelity) relative to  $\lambda_1$  (image quality), forcing the encoder to prioritise robustness over imperceptibility until performance stabilises. This curriculum-style approach avoids the common failure mode in which the encoder converges to an imperceptible but fragile embedding from which it cannot escape by small gradient steps.

**Reported performance.** ReDMark achieves a PSNR of 36.2 dB at 30-bit payload with a BER of 1.4% after JPEG compression at quality factor 50, and remains below 5% BER under moderate rotation ( $\leq 3$ ) and scaling ( $\leq 10\%$ ). These results represent a meaningful advance over HiDDeN and SteganoGAN under the same attack conditions.

**Limitations.** Despite these improvements, ReDMark retains the fundamental distribution-shift vulnerability. Its resistance to geometric attacks extends only to the mild transformations simulated during training. Under large-scale cropping (removal of  $> 30\%$  of image content) or compound attacks combining JPEG compression with significant resizing (e.g. the pipeline applied by Twitter and WhatsApp), the BER rises sharply. Furthermore, ReDMark, like all decoder-only deep methods, provides no mechanism for partial recovery: if the bit error rate exceeds the decoder’s correction capability, the output is unintelligible, and no information about which fragments of the payload survived is available.

### 2.6.5 Hiding Images in Images: High-Capacity Deep Concealment

A parallel line of deep steganographic research is concerned not with hiding binary messages but with concealing entire images within other images. Baluja [38] introduced a CNN-based framework for hiding a secret image within a cover image of the same dimensions, using a “preparation” network

to transform the secret image into a residual-ready representation before the hiding network embeds it into the cover. The reveal network then reconstructs the secret image from the stego-image with high fidelity (PSNR above 30 dB for both the stego-image and the revealed secret).

Lu et al. [47] extended this paradigm to *large-capacity* image hiding, proposing an invertible neural network (INN) architecture [48] that is mathematically reversible: the same network parameters are used for both embedding and extraction by exploiting the invertibility of normalising flows. This guarantees that in the absence of any attack, perfect recovery is possible. Under compression attacks, however, the invertibility is broken, and recovery degrades.

While the image-in-image paradigm is not directly applicable to the binary message concealment task of this thesis, it illustrates the general capacity of deep architectures to learn complex, content-adaptive embeddings that far exceed the capacity of classical transform-domain methods.

### 2.6.6 Robust Deep Steganography Against Social Network Processing

A practically important research direction targets the specific degradation pipelines applied by social media platforms (Instagram, Twitter, WhatsApp, Facebook), which combine JPEG recompression, resolution rescaling, colour-space conversion, and sometimes format conversion (e.g. JPEG to WebP). These operations collectively remove a large fraction of the information embedded by conventional and naive deep steganographic methods.

Wengrowski and Dana [49] proposed a differentiable model of the photographic process (capture, print, display, and re-capture), enabling a neural steganographic system to survive the full print-scan loop. Their system produces cover-looking images that can be photographed with a mobile camera and decoded with the neural decoder, surviving both compression and optical blur.

Jia et al. [50] introduced MBRS (*Mini-Batch of Real and Simulated JPEG Compression*), in which each training mini-batch contains stego-images processed by both a differentiable JPEG simulation and a *real* JPEG codec. Including real JPEG artifacts in the training data closes the gap between the differentiable approximation and the actual codec behaviour, yielding substantially lower BER under real-world JPEG compression compared to methods trained on differentiable approximations alone. MBRS achieves 0.9% BER at 30-bit payload under JPEG quality factor 50 and 1.3% under quality factor 30, with a stego-image PSNR of 38.1 dB.

### 2.6.7 Attention Mechanisms and Semantic-Aware Embedding

Transformer-based architectures and spatial attention mechanisms have been applied to steganography to improve both imperceptibility and robustness by enabling the network to focus embedding on semantically stable image regions.

Luo et al. [51] proposed a *distortion-agnostic* deep hiding network that uses a channel-spatial attention module to estimate the local imperceptibility budget at each pixel, concentrating embedding in texture-rich regions (similar to the JND-based adaptive embedding discussed in Section 2.3.4) while learning the content of those regions from data rather than from hand-crafted JND models. The attention weights are updated jointly with the encoder and decoder weights during training, allowing the system to adapt its embedding map to the statistics of natural images.

Zhang et al. [52] introduced UDH (*Universal Deep Hiding*), a framework in which the encoder learns a single, universal embedding pattern that is content-independent, i.e. the same residual signal is added to any cover image regardless of its content. While this reduces capacity compared to content-adaptive methods, it makes the embedding pattern universal and more predictable under attacks, which simplifies decoder design.

### 2.6.8 Hybrid Transform-Network Approaches

A natural development in the field is the combination of classical frequency-domain transforms with learned components, seeking to obtain the theoretical guarantees and interpretability of transform-

domain methods alongside the adaptivity of data-driven approaches.

Ahmadi et al. [53] proposed a framework in which the encoder operates in the DCT domain: DCT coefficients are first extracted from the cover image, and the neural network learns to modify a selected subset of mid-frequency coefficients as a function of both the cover content and the secret message. The extraction network inverts this process in the DCT domain. This hybrid approach inherits the JPEG-robustness advantage of DCT embedding and the adaptivity advantage of learned coefficient selection, achieving PSNR of 40.1 dB and BER of 2.1% under JPEG quality factor 70.

Fernandez et al. [54] explored the integration of steganography with latent diffusion models, embedding messages in the latent space of a pre-trained image generator. This approach exploits the perceptual quality of modern generative models to produce stego-images that are virtually indistinguishable from natural photographs, at the cost of requiring a generative model at inference time and being restricted to the image manifold of the training data.

### 2.6.9 Deep Learning Steganalysis and Its Impact on Robust Design

The same deep learning revolution that has advanced steganographic embedding has also produced substantially more powerful steganalytic detectors. The SRNet architecture of Boroumand et al. [55] achieves near-perfect detection of several state-of-the-art spatial-domain and DCT-domain steganographic algorithms using a single, unified deep network, without any algorithm-specific feature engineering. More recently, the XuNet [56] and Ye-Net [57] architectures have demonstrated that deep classifiers trained on the StegoAppDB dataset generalise across embedding algorithms, significantly raising the bar for undetectability.

These developments have important implications for the design of robust steganographic systems. Because deep steganalysis is both universal (algorithm-agnostic) and sensitive (detecting payloads at embedding rates well below 0.1 bpp in spatial-domain methods), any strategy that increases embedding strength for the sake of robustness automatically increases the statistical footprint detectable by a deep steganalyser. This confirms and deepens the detectability–robustness trade-off discussed in Section 2.3.7, and reinforces the motivation for structural robustness approaches (fragmentation, erasure coding, domain diversity) that can maintain a low per-pixel distortion while still enabling payload survivability.

### 2.6.10 Summary Table and Comparative Analysis

Table 2.2 summarises the deep learning-based methods reviewed in this section alongside the classical methods from Section 2.3, providing a unified comparative view.

The critical observation from Table 2.2 is consistent with the finding of Table 2.1 in Section 2.3.9: no existing deep learning system, including the most recent state-of-the-art methods, provides mechanisms for *partial recovery* of the hidden payload when large portions of the stego-image are destroyed. All deep decoders operate in a binary succeed-or-fail mode: they either reconstruct the entire message or produce unintelligible output. This is a fundamental architectural limitation of the encoder-decoder paradigm, not a parameter that can be tuned.

### 2.6.11 Why This Thesis Adopts a Classical Framework

The review in this section makes clear that deep learning-based steganography represents the current state of the art in terms of raw imperceptibility and robustness to trained attacks. Nevertheless, the present thesis adopts a classical, model-driven framework for several reasons that are directly related to the specific research objectives defined in Chapter 1.

**Partial recovery as a first-class design objective.** The primary objective of this thesis is to design a system that can recover *usable information* from a partially destroyed stego-image, rather than a system that achieves the lowest BER under a fixed set of trained attacks. As established

Table 2.2: Comparative summary of deep learning-based steganographic methods (2018–2024) alongside selected classical baselines. Performance figures are indicative only and refer to 30-bit payloads on natural image datasets under JPEG QF = 70 unless otherwise noted. BER: Bit Error Rate; PSNR: Peak Signal-to-Noise Ratio.

Method	Year	PSNR (dB)	BER (%)	Geo. Rob.	Partial Rec.
DCT mid-freq. (classical)	—	≈38	≈15	Poor	No
LSB spatial (classical)	—	≈51	≈50	Poor	No
HiDDeN [35]	2018	33.1	0.5	Limited	No
SteganoGAN [41]	2019	30.5	40+ (JPEG)	None	No
ReDMark [43]	2021	36.2	1.4	Limited	No
MBRS [50]	2021	38.1	0.9	Limited	No
UDH [52]	2020	34.8	2.2	Limited	No
DCT-hybrid [53]	2020	40.1	2.1	Limited	No
<b>Proposed framework</b>	2025	—	—	Designed	<b>Yes</b>

above, no existing deep learning framework provides partial recovery capabilities; the encoder-decoder paradigm produces a holistic encoding of the entire payload, and the decoder requires the full encoded signal to function. Classical fragmentation and erasure-coding strategies (Section 2.3.5) are directly applicable to the partial-recovery objective in a way that is not yet achievable with deep architectures.

**Robustness to *unseen* compound attacks.** Deep steganographic systems are robust only to the attacks represented in the training noise layer. Social media platforms change their processing pipelines frequently, and different platforms apply different combinations of compression quality, resize ratios, and format conversions. A classical system whose robustness derives from structural properties of the payload (redundancy, distributed embedding, erasure coding) is robust by construction to any attack that does not exceed its erasure budget, regardless of whether that specific attack was anticipated during design. This generalisation property is especially important for deployment in uncontrolled environments.

**Interpretability and analytical robustness guarantees.** A classical steganographic framework admits formal analysis: given a specified attack model, one can compute (or bound) the probability that a given fragment survives, and hence the probability that the payload is recovered. Deep learning systems provide no such guarantees; their robustness is empirical and may degrade in ways that are difficult to predict. For applications in which reliability is a hard requirement, interpretability and analytical tractability are significant advantages.

**Computational accessibility.** Training a competitive deep steganographic system requires large-scale image datasets (typically tens of thousands of images) and GPU-accelerated computing infrastructure for extended training runs (often 24–100 hours). The classical framework proposed in this thesis is fully implementable on standard hardware without any training phase, making it accessible in resource-constrained environments.

**Modularity and domain expertise.** The classical framework allows individual components (the ECC scheme, the fragmentation strategy, the embedding domain, the embedding strength) to be adjusted independently based on the properties of the specific deployment environment. In a deep system, these design choices are entangled within the network weights and cannot be modified without

retraining. This modularity is particularly valuable when the attack model is only partially known in advance.

In summary, this thesis positions itself within the classical stream of robust steganography research, but draws inspiration from the structural insights of deep learning approaches — in particular, the idea that robustness is best achieved through a system-level design philosophy rather than through individually robust components. The proposed framework combines structured payload fragmentation, multi-domain embedding, and erasure-code redundancy to achieve the partial-recovery objective that neither classical single-domain methods nor current deep learning systems have addressed.

## 2.7 Evaluation Criteria in Steganographic Systems

The design and experimental evaluation of any steganographic system requires a precise set of quantitative metrics that measure performance along the three fundamental axes introduced in Section 2.1.1: imperceptibility, robustness, and (where applicable) steganalysis resistance. A clear understanding of these metrics — their mathematical definitions, their practical interpretation, and their limitations — is essential both for comparing different steganographic methods in the literature and for assessing the proposed framework in Chapter 5.

This section provides a comprehensive treatment of the evaluation criteria used throughout this thesis, divided into two groups: imperceptibility metrics (Section 2.7.1) and robustness metrics (Section 2.7.2).

### 2.7.1 Imperceptibility Metrics

Imperceptibility metrics quantify the visual and statistical difference between a cover image and the corresponding stego-image. A high-quality embedding is one in which these differences are small enough to be undetectable by human observers and difficult to distinguish statistically from natural image noise. The three metrics used in this work are the Mean Squared Error (MSE), the Peak Signal-to-Noise Ratio (PSNR), and the Structural Similarity Index Measure (SSIM).

#### Mean Squared Error (MSE)

The Mean Squared Error is the most fundamental measure of pixel-level distortion between two images. Given a cover image  $C$  and a stego-image  $S$ , each of dimensions  $M \times N$  pixels, the MSE is defined as:

$$\text{MSE}(C, S) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [C(i, j) - S(i, j)]^2, \quad (2.21)$$

where  $C(i, j)$  and  $S(i, j)$  denote the intensity values of the cover and stego-images at pixel location  $(i, j)$ , respectively. For colour images, the MSE is commonly computed channel-wise and averaged:

$$\text{MSE}_{\text{colour}} = \frac{1}{3} (\text{MSE}_R + \text{MSE}_G + \text{MSE}_B). \quad (2.22)$$

The MSE is measured in squared intensity units (e.g. squared grey levels for 8-bit images, 0–255<sup>2</sup>). A lower MSE value indicates less pixel-level distortion. However, MSE has a well-known limitation: equal MSE values can correspond to perceptually very different types of distortion. For example, a small uniform shift in pixel values across a flat region produces the same MSE as an equal-energy pattern of localised noise, but the latter is far more visually conspicuous. For this reason, MSE is rarely used as the sole measure of imperceptibility in steganographic evaluations; it serves primarily as the mathematical foundation for PSNR.

## Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio is the most widely used imperceptibility metric in the steganography and image processing literature [2]. It expresses the ratio of the maximum possible pixel energy to the distortion energy introduced by embedding, measured on a logarithmic (decibel) scale:

$$\text{PSNR}(C, S) = 10 \cdot \log_{10} \left( \frac{L^2}{\text{MSE}(C, S)} \right) \quad [\text{dB}], \quad (2.23)$$

where  $L$  is the maximum possible pixel intensity value. For 8-bit images,  $L = 255$ ; for 16-bit images,  $L = 65535$ . The logarithmic scale is chosen because human perception of brightness differences is approximately logarithmic. PSNR is always expressed in decibels (dB).

**Interpretation and thresholds.** PSNR is a monotonically increasing function of perceptual quality. The following thresholds serve as practical guidelines in the steganographic literature [8]:

- $\text{PSNR} > 40 \text{ dB}$ : Excellent imperceptibility; the stego-image is typically indistinguishable from the cover image to human observers under normal viewing conditions.
- $35 \leq \text{PSNR} \leq 40 \text{ dB}$ : Good imperceptibility; the embedding is acceptable for most applications where human inspection is not performed under magnification.
- $30 \leq \text{PSNR} < 35 \text{ dB}$ : Acceptable imperceptibility; a trained observer may detect subtle artefacts, but casual inspection will not reveal the embedding.
- $\text{PSNR} < 30 \text{ dB}$ : Poor imperceptibility; visible artefacts are present and the embedding is unsuitable for covert communication.

**Limitations.** Despite its ubiquity, PSNR has well-documented limitations as a perceptual quality metric [58]. It treats all pixel positions and all spatial frequencies as equally important, whereas the human visual system (HVS) is significantly more sensitive to distortion in smooth, homogeneous image regions than in textured areas, and more sensitive to low-frequency errors than to high-frequency ones. Two stego-images with identical PSNR values may therefore be perceived very differently by a human observer. These limitations motivate the use of SSIM as a complementary metric.

**Relationship between PSNR and embedding capacity.** For a steganographic method that embeds data by adding an embedding signal of energy  $E$  to the cover image, Equation (2.23) shows that PSNR decreases by approximately 3 dB for every doubling of the embedding strength. More precisely, for a fixed image size  $MN$  and a fixed embedding signal, doubling the number of payload bits requires doubling the average energy per bit, reducing the PSNR by 3 dB. This energy-capacity relationship is the mathematical expression of the imperceptibility–capacity trade-off discussed in Section 2.1.1.

## Structural Similarity Index Measure (SSIM)

The Structural Similarity Index Measure was introduced by Wang et al. [58] as a perceptual quality metric that models the sensitivity of the HVS to structural information. The key insight is that natural images are highly structured: neighbouring pixels are strongly correlated, and these correlations carry important perceptual information about object boundaries, textures, and surfaces. SSIM measures the degradation of this structural information rather than pixel-level intensity differences.

SSIM is computed locally over rectangular windows (typically  $11 \times 11$  pixels with a Gaussian weighting function) and then averaged across all window positions to produce a single global score.

For two image patches  $\mathbf{x}$  and  $\mathbf{y}$  drawn from the cover and stego-images respectively, the local SSIM is defined as:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \underbrace{\frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}}_{\text{luminance}} \cdot \underbrace{\frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}}_{\text{contrast}} \cdot \underbrace{\frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}}_{\text{structure}}, \quad (2.24)$$

where  $\mu_x$ ,  $\mu_y$  are the mean intensities of the two patches;  $\sigma_x^2$ ,  $\sigma_y^2$  are their variances;  $\sigma_{xy}$  is their covariance; and  $c_1$ ,  $c_2$ ,  $c_3$  are small stabilisation constants that prevent division by zero (typically  $c_1 = (0.01L)^2$ ,  $c_2 = (0.03L)^2$ ,  $c_3 = c_2/2$ ).

**Interpretation.** The SSIM index ranges from  $-1$  to  $1$ , where  $1$  indicates perfect structural similarity (identical images) and  $0$  indicates no structural correlation. In practice, for high-quality stego-images, SSIM values are very close to  $1$ :

- $\text{SSIM} > 0.99$ : Excellent; structural differences are imperceptible.
- $0.97 \leq \text{SSIM} \leq 0.99$ : Good; minor structural changes, not visible under normal conditions.
- $0.90 \leq \text{SSIM} < 0.97$ : Acceptable; some structural artefacts may be visible in textured regions.
- $\text{SSIM} < 0.90$ : Poor; structural degradation is perceptible.

SSIM is generally considered a more reliable predictor of perceived image quality than PSNR, particularly for compression-type distortions [58]. It is therefore reported alongside PSNR in the experimental evaluation of Chapter 5.

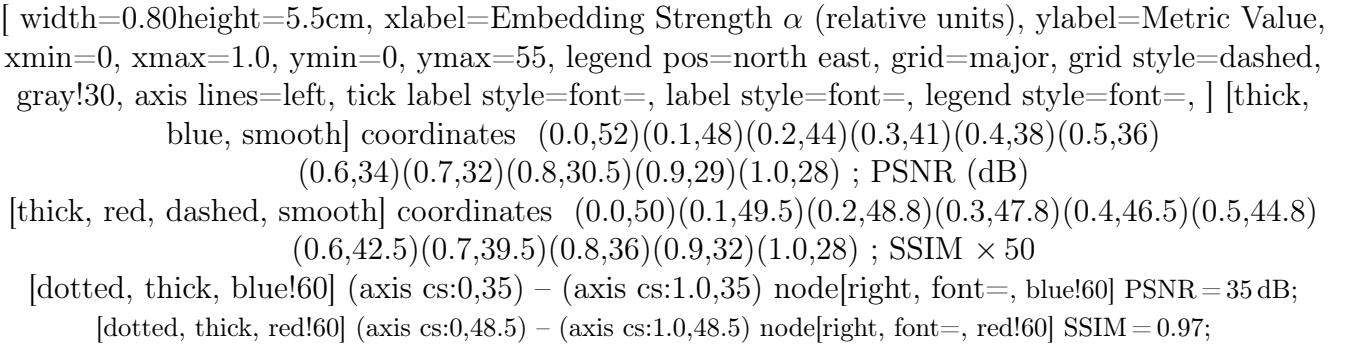


Figure 2.9: Illustrative relationship between embedding strength  $\alpha$  and the two imperceptibility metrics PSNR and SSIM (scaled by 50). Both metrics decrease monotonically with increasing embedding strength. PSNR degrades rapidly even at moderate strengths, while SSIM is more tolerant of small-amplitude, high-frequency embedding signals that preserve structural content. This highlights the complementary nature of the two metrics.

## Relationship Between Imperceptibility Metrics

Although PSNR and SSIM are both imperceptibility metrics, they capture different aspects of image quality and can sometimes disagree on the ranking of steganographic methods. Embedding in the low-frequency components of an image (e.g. the DC coefficients of DCT) produces large intensity shifts that severely degrade PSNR but may preserve structural correlations and therefore maintain a relatively high SSIM. Conversely, high-frequency noise-like embeddings may produce a high PSNR (because energy is spread thinly across many pixels) while destroying textural structures and yielding a low SSIM. This complementarity explains why both metrics are reported in Chapter 5 and why neither alone provides a complete picture of imperceptibility.

## 2.7.2 Robustness Metrics

Robustness metrics quantify the ability of a steganographic system to recover the hidden message after the stego-image has been subjected to signal processing or geometric attacks. The two metrics used in this work are the Bit Error Rate (BER) and the Normalised Correlation (NC).

### Bit Error Rate (BER)

The Bit Error Rate is the primary robustness metric in steganography and digital communications [3]. It measures the fraction of payload bits that are incorrectly recovered after extraction from a (possibly degraded) stego-image. Formally, let  $M = [m_1, m_2, \dots, m_K]$  be the original binary message of length  $K$  bits, and let  $\hat{M} = [\hat{m}_1, \hat{m}_2, \dots, \hat{m}_K]$  be the extracted message after attack. The BER is defined as:

$$\text{BER}(M, \hat{M}) = \frac{1}{K} \sum_{k=1}^K \mathbf{1}[\hat{m}_k \neq m_k], \quad (2.25)$$

where  $\mathbf{1}[\cdot]$  is the indicator function that equals 1 if its argument is true and 0 otherwise. The BER is a dimensionless quantity in the range  $[0, 1]$ , typically expressed as a percentage or in scientific notation.

### Interpretation and thresholds.

- $\text{BER} = 0.0$  (or 0%): Perfect recovery; every bit of the hidden message is correctly extracted. This is the ideal outcome and is achievable only in the absence of attacks or with sufficiently powerful error correction coding.
- $\text{BER} \leq 0.01$  ( $\leq 1\%$ ): Excellent robustness; with appropriate error correction coding, the original message can be recovered reliably.
- $\text{BER} \leq 0.05$  ( $\leq 5\%$ ): Good robustness; recovery is possible with moderate ECC overhead.
- $\text{BER} \approx 0.5$  ( $\approx 50\%$ ): Complete failure; the extracted bits are no better than random, meaning the embedding has been entirely destroyed by the attack. This is the expected BER when attempting to extract a message from a cover image in which nothing was embedded.

**Relationship to error correction coding.** When the steganographic payload is protected by an error correction code with correction capacity  $t$  (i.e. the code can correct up to  $t$  bit errors per codeword), perfect message recovery is possible if and only if the BER satisfies:

$$\text{BER} < \frac{t}{n}, \quad (2.26)$$

where  $n$  is the codeword length. For example, a Reed-Solomon code with  $n = 255$  and  $t = 31$  can correct up to  $\approx 12.2\%$  BER. This relationship motivates the use of ECC in the proposed framework and defines the BER budget available to the embedding and transmission process.

**BER under partial data loss.** In the context of the present thesis, where the stego-image may be subjected to cropping or localised tampering, the effective BER includes contributions from two distinct sources: (i) signal-level errors caused by noise and compression in the surviving image regions, and (ii) erasure errors caused by the complete removal of image regions containing embedded fragments. These two error types require different mitigation strategies — ECC for signal-level errors and erasure codes or redundancy for erasure errors — and are therefore tracked separately in the experimental evaluation of Chapter 5.

## Normalised Correlation (NC)

The Normalised Correlation provides an alternative measure of message recovery fidelity that is more robust to systematic bit inversions than the raw BER. It is commonly used in digital watermarking [8] and has been adopted in robust steganographic evaluation to provide a continuous measure of recovery quality that degrades smoothly as attack intensity increases. The NC between the original message  $M$  and the extracted message  $\hat{M}$ , after bipolar encoding ( $0 \rightarrow -1$ ,  $1 \rightarrow +1$ ), is defined as:

$$\text{NC}(M, \hat{M}) = \frac{1}{K} \sum_{k=1}^K \bar{m}_k \cdot \hat{\bar{m}}_k, \quad (2.27)$$

where  $\bar{m}_k \in \{-1, +1\}$  and  $\hat{\bar{m}}_k \in \{-1, +1\}$  are the bipolarly encoded versions of the original and extracted bits, respectively.

### Interpretation.

- $\text{NC} = 1.0$ : Perfect recovery; all bits agree.
- $\text{NC} = 0.0$ : Complete failure; extracted bits are uncorrelated with the original, as would occur if a random sequence were substituted.
- $\text{NC} = -1.0$ : All bits are inverted; this can occur under systematic polarity reversal attacks and would be incorrectly reported as  $\text{BER} = 1.0$  (perfect failure) even though the message is recoverable by simple inversion. The NC captures this case.

The relationship between NC and BER for independently and identically distributed bit errors is:

$$\text{NC} = 1 - 2 \cdot \text{BER}, \quad (2.28)$$

from which it follows that  $\text{BER} = 0$  corresponds to  $\text{NC} = 1$  and  $\text{BER} = 0.5$  corresponds to  $\text{NC} = 0$ . Both metrics therefore convey the same information for independent errors, but NC is preferable when systematic errors or polarity reversals are possible. In this work, both metrics are reported to facilitate comparison with results from the watermarking literature, which predominantly uses NC.

### 2.7.3 Partial Recovery Rate (PRR)

Classical steganographic evaluation is concerned with binary outcomes: either the entire message is recovered ( $\text{BER} \approx 0$ ) or it is not. In the context of the self-recovering framework proposed in this thesis, however, the system is explicitly designed to recover *partial* messages when not all embedded fragments survive. A metric that captures this partial-recovery behaviour is therefore needed.

We define the **Partial Recovery Rate (PRR)** as the fraction of payload fragments that are successfully recovered (with BER below a threshold  $\epsilon$  for each fragment) from the degraded stego-image:

$$\text{PRR} = \frac{|\{i : \text{BER}(F_i, \hat{F}_i) \leq \epsilon\}|}{N_F}, \quad (2.29)$$

where  $N_F$  is the total number of fragments in the structured payload,  $F_i$  is the  $i$ -th original fragment,  $\hat{F}_i$  is the extracted version, and  $\epsilon$  is a per-fragment BER threshold (typically 0.05 when ECC is applied at the fragment level). A PRR of 1.0 indicates complete recovery; a PRR of 0.5 indicates that half the message fragments were correctly extracted; a PRR of 0.0 indicates total failure. This metric is unique to the present work and is reported in Chapter 5 to assess the partial-recovery performance of the proposed framework under cropping and localised tampering attacks.

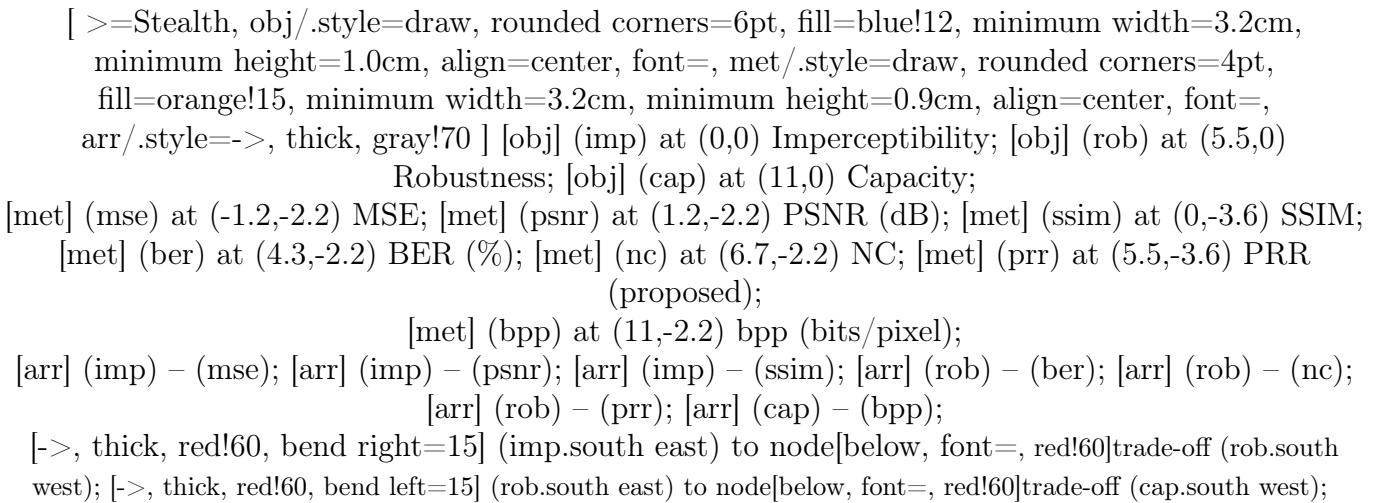


Figure 2.10: Relationship between the three steganographic objectives and the quantitative evaluation metrics used in this thesis. The trade-offs between objectives are captured by tracking all metrics simultaneously across experimental conditions. PRR (Partial Recovery Rate) is a metric introduced in this work to assess the unique partial-recovery capability of the proposed framework.

## 2.7.4 Unified View of Evaluation Criteria

Figure 2.10 provides a visual overview of the relationship between the evaluation metrics defined in this section and the three steganographic objectives.

Table 2.3 summarises all evaluation metrics, their definitions, their measurement units, and the performance thresholds adopted in this thesis.

Table 2.3: Summary of evaluation metrics used in this thesis. Thresholds represent commonly accepted standards in the steganographic and image quality literature.

Metric	Objective	Formula	Range	Good threshold
MSE	Imperceptibility	$\frac{1}{MN} \sum (C - S)^2$	$[0, \infty)$	As small as possible
PSNR	Imperceptibility	$10 \log_{10}(L^2 / \text{MSE})$	(0, $\infty$ ) dB	> 35 dB
SSIM	Imperceptibility	Eq. (2.24)	$[-1, 1]$	> 0.97
BER	Robustness	$\frac{1}{K} \sum \mathbf{1}[\hat{m} \neq m]$	$[0, 1]$	< 0.05
NC	Robustness	$\frac{1}{K} \sum \bar{m} \cdot \hat{\bar{m}}$	$[-1, 1]$	> 0.9
PRR	Partial recovery	Eq. (2.29)	$[0, 1]$	> 0.7 (proposed)
bpp	Capacity	bits embedded per pixel	(0, 8]	Application-dependent

These metrics, taken together, provide a comprehensive and multi-dimensional assessment of steganographic performance. In Chapter 5, each metric is reported for the proposed framework and for the classical baselines under each attack scenario, enabling a rigorous comparative evaluation that respects the multi-objective nature of the steganographic design problem.

# Chapter 3

## Proposed Self-Recovering Steganography Framework

### 3.1 Design Motivation and Objectives

This chapter introduces the proposed steganographic framework, which is designed to address the limitations identified in the literature review, particularly the fragility of conventional embedding methods under severe image degradation and geometric desynchronization.

The primary design goal of the proposed framework is to enhance payload survivability rather than solely reinforcing embedding locations. Unlike traditional approaches that assume intact data and perfect synchronization, the proposed system explicitly assumes partial data loss and uncontrolled image transformations. The framework is therefore designed to tolerate damage, enable partial recovery, and reconstruct missing information using structured payload relationships.

### 3.2 High-Level System Overview

This section presents a conceptual overview of the proposed framework, illustrating the main processing stages at both the sender and receiver sides.

At a high level, the system consists of four main components:

- Payload structuring and self-description
- Fragmentation and inter-fragment relationship encoding
- Multi-domain embedding strategy
- Damage-aware extraction and reconstruction

The interaction between these components enables robust message recovery even when a significant portion of the embedded data is degraded or removed.

#### 3.2.1 Sender-Side Processing Pipeline

The sender-side pipeline begins with the secret message and produces a stego-image suitable for transmission through uncontrolled digital environments.

The main stages include:

- Conversion of the secret message into a structured payload
- Fragment generation and relationship encoding
- Selection of embedding domains and regions
- Embedding of fragments into the cover image

### 3.2.2 Receiver-Side Processing Pipeline

The receiver-side pipeline operates on a potentially degraded stego-image and attempts to recover the original message.

The extraction process includes:

- Detection and extraction of embedded fragments
- Damage estimation and fragment confidence evaluation
- Fragment validation and relationship checking
- Partial reconstruction and message reassembly

## 3.3 Structured Payload Representation

Instead of embedding raw binary data, the proposed framework transforms the secret message into a self-describing structured payload. This payload contains not only the original message bits but also additional information that enables validation, ordering, and reconstruction.

### 3.3.1 Payload Components

Each payload fragment contains the following elements:

- Fragment identifier and ordering information
- Local payload data
- Lightweight checksum for integrity verification
- Cross-fragment reference information

This structure allows individual fragments to contribute to the reconstruction of missing or corrupted fragments.

### 3.3.2 Inter-Fragment Relationships

Fragments are not treated as independent entities. Instead, each fragment stores partial information about neighboring and non-neighboring fragments. These relationships form a redundancy graph that supports reconstruction under partial data loss.

The use of relational information enables recovery even when contiguous regions of the image are destroyed.

## 3.4 Fragmentation Strategy

The structured payload is divided into multiple interconnected fragments to increase resilience against localized damage.

### 3.4.1 Fragment Size and Granularity

Fragment size is chosen as a compromise between robustness and capacity. Smaller fragments improve survivability under cropping, while larger fragments reduce overhead.

This work adopts a fixed fragment size to simplify reconstruction and evaluation.

### **3.4.2 Redundancy and Distribution Policy**

Controlled redundancy is introduced at the fragment level rather than the bit level. Each fragment is embedded multiple times across different domains or image regions according to a predefined distribution policy.

The redundancy level is configurable and directly impacts robustness and payload capacity.

## **3.5 Multi-Domain Embedding Strategy**

To reduce vulnerability to specific attack types, fragments are embedded across multiple embedding domains, each offering resistance to different forms of degradation.

### **3.5.1 Spatial Domain Embedding**

Fragments embedded in spatial texture regions are resilient to localized cropping and partial tampering. Texture-based masking is used to minimize visual distortion.

### **3.5.2 Frequency Domain Embedding**

Mid-frequency DCT coefficients are used to embed fragments resistant to JPEG compression and moderate noise. Embedding strength is adjusted to balance imperceptibility and robustness.

### **3.5.3 Multi-Scale and Redundant Embedding**

Selected fragments are embedded at different spatial resolutions or downsampled representations of the image to improve resilience against resizing operations.

### **3.5.4 Color Channel Utilization**

For color images, redundancy is introduced across luminance and chrominance channels. Stronger embedding is applied to the luminance channel, while weaker redundant embedding is used in chrominance channels.

## **3.6 Embedding Control and Parameter Selection**

This section describes the parameters governing embedding strength, fragment placement, and redundancy levels.

### **3.6.1 Key-Based Fragment Placement**

A secret key is used to pseudo-randomly determine fragment locations and embedding domains, enhancing security and preventing unauthorized extraction.

### **3.6.2 Embedding Strength Adaptation**

Embedding strength is adjusted based on local image characteristics, such as texture intensity and frequency stability.

## 3.7 Damage-Aware Extraction Process

Unlike traditional extraction schemes, the proposed framework does not assume intact data. Instead, it explicitly estimates damage and adapts the extraction process accordingly.

### 3.7.1 Fragment Detection and Validation

Extracted fragments are validated using checksums and structural consistency checks. Invalid or severely corrupted fragments are discarded or assigned low confidence.

### 3.7.2 Confidence Scoring Mechanism

Each fragment is assigned a confidence score based on extraction quality, consistency with neighboring fragments, and agreement with relational information.

### 3.7.3 Adaptive Fragment Selection

Fragments with higher confidence scores are prioritized during reconstruction. Lower-confidence fragments are used only when necessary.

## 3.8 Message Reconstruction and Self-Recovery

The final message is reconstructed using surviving fragments and their encoded relationships.

### 3.8.1 Partial Reconstruction Strategy

Missing fragments are reconstructed using relational summaries and redundancy information provided by neighboring fragments.

### 3.8.2 Failure Conditions

This section defines conditions under which full or partial recovery is not possible, such as excessive data loss or severe compound attacks.

## 3.9 Computational Complexity and Practical Considerations

The computational overhead of the proposed framework is analyzed in terms of embedding time, extraction time, and memory requirements.

### 3.9.1 Complexity Analysis

The complexity of payload structuring, embedding, and extraction is discussed qualitatively.

### 3.9.2 Implementation Constraints

Practical limitations related to image size, payload capacity, and real-world deployment are identified.

### **3.10 Summary of the Proposed Framework**

This section summarizes the key design principles and highlights how the proposed framework addresses the research gaps identified in Chapter 2, particularly synchronization loss, partial data destruction, and robustness under compound attacks.

# Chapter 4

## Implementation smdljkf and Experimental Setup

### 4.1 Development Environment

The tools and software environment used for implementation are described.

### 4.2 Dataset Description

The image datasets used for experimentation are presented.

### 4.3 Attack Simulation

This section describes how image degradation and tampering scenarios are simulated.

### 4.4 Evaluation Metrics

#### 4.4.1 Imperceptibility Metrics

PSNR and SSIM are introduced.

#### 4.4.2 Robustness Metrics

Bit Error Rate (BER) and Normalized Correlation (NC) are defined.

# **Chapter 5**

## **Results and Discussion**

### **5.1 Imperceptibility Evaluation**

Visual quality and distortion analysis are presented.

### **5.2 Robustness Under Image Degradation**

Performance under compression, noise, and resizing is analyzed.

### **5.3 Robustness Under Partial Data Loss**

The impact of cropping and localized tampering is evaluated.

### **5.4 Comparative Analysis**

The proposed framework is compared with classical steganography methods.

### **5.5 Discussion of Results**

Strengths, weaknesses, and observed trade-offs are discussed.

# **Chapter 6**

## **Conclusion and Future Work**

### **6.1 Summary of Findings**

The main outcomes of the study are summarized.

### **6.2 Limitations**

Practical and theoretical limitations of the framework are discussed.

### **6.3 Future Research Directions**

Possible extensions and improvements are suggested.

# Bibliography

- [1] T. Moerland. Steganography and steganalysis. Technical Report 1, Leiden Institute of Advanced Computing Science, 2005. Available at: <https://www.liacs.nl/home/tmoerl/privtech.pdf>.
- [2] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.
- [3] John G. Proakis and Masoud Salehi. *Digital Communications*. McGraw-Hill, New York, NY, USA, 5th edition, 2008.
- [4] Tayana Morkel, Jan HP Elof, and Martin S Olivier. An overview of image steganography. Number 2, 2005.
- [5] Jessica Fridrich and Jan Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
- [6] Niels Provos and Peter Honeyman. Hide and seek: An introduction to steganography. In *IEEE Security & Privacy*, pages 32–44, 2003.
- [7] Neil F. Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *IEEE Computer*, 31(2):26–34, 1998.
- [8] Ingemar Cox, Matthew Miller, and Jeffrey Bloom. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.
- [9] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding—a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.
- [10] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [11] Po-Yueh Chen and Hung-Ju Lin. A dwt based approach for image steganography. *International Journal of Applied Science and Engineering*, 4(3):275–290, 2006.
- [12] Gregory K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.
- [13] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
- [14] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, 2000.
- [15] J. J. K. Ó Ruanaidh and T. Pun. Rotation, scale and translation invariant digital image watermarking. In *IEEE International Conference on Image Processing*, 1998.
- [16] V. Solachidis and I. Pitas. Circularly symmetric watermark embedding in 2-d dft domain. In *IEEE Transactions on Image Processing*, volume 10, pages 1741–1753, 2001.
- [17] Lisa M. Marvel, Charles G. Boncelet, and Charles T. Retter. Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8(8):1075–1083, 1999.

- [18] Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, pages 529–541, 1981.
- [19] Peter Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.
- [20] I. Pitas. A method for signature casting on digital images. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 215–218, 1996.
- [21] Fredric J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.
- [22] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [23] Christine I. Podilchuk and Wenjun Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, 1998.
- [24] Brian Chen and Gregory W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001.
- [25] M. Barni, F. Bartolini, V. Cappellini, and A. Piva. A dct-domain system for robust image watermarking. *Signal Processing*, 66(3):357–372, 1998.
- [26] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, 1999.
- [27] Martin Kutter. Digital signature of color images using amplitude modulation. In *Storage and Retrieval for Image and Video Databases*, volume 3022, pages 518–526, 1999.
- [28] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2nd edition, 2007.
- [29] Gérard Petit. Pour une conception lexicologique de la dénomination. *Cahiers de praxématique*, (36):93–115, 2001.
- [30] Joseph J. K. Ó Ruanaidh and Thierry Pun. *Digital Watermarking*. Springer, 1998.
- [31] Joseph J. K. Ó Ruanaidh and Thierry Pun. Rotation, scale and translation invariant digital image watermarking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 536–539. IEEE, 1998.
- [32] Mauro Barni, Franco Bartolini, and Alessandro Piva. Multichannel watermarking of color images. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):142–156, 2002.
- [33] Neil F. Johnson and Sushil Jajodia. Steganography: Seeing the unseen. In *IEEE Computer*, pages 26–34, 1998.
- [34] Niels Provos and Peter Honeyman. Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, 1(3):32–44, 2003.
- [35] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 682–697, Cham, Switzerland, 2018. Springer.
- [36] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*, 2017.

- [37] Jian Ye, Jiangqun Ni, Yang Yi, and Dengpan Ye. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.
- [38] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- [39] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.
- [40] Andreas Westfeld. F5 — a steganographic algorithm: High capacity despite better steganalysis. pages 289–302, 2001.
- [41] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. SteganoGAN: High capacity image steganography with GANs. In *arXiv preprint arXiv:1901.03892*, 2019.
- [42] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.
- [43] Souvik Das, Chia-Yu Lin, and Hung-Yu Wei. ReDMark: Framework for residual diffusion watermarking based on deep neural networks. In *Expert Systems with Applications*, volume 166, page 114085. Elsevier, 2021.
- [44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.
- [45] Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017.
- [46] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.
- [47] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33:10223–10234, 2020.
- [48] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real-valued Non-Volume Preserving (Real NVP) transformations*. 2017.
- [49] Eric Wengrowski and Kristin Dana. Light field messaging with deep photographic steganography. pages 1515–1524, 2019.
- [50] Zhaoyang Jia, Han Fang, and Weiming Zhang. MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression. In *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*, pages 41–49, 2021.
- [51] Xiyang Luo, Ruohan Zhan, Chi-Hao Chang, Feng Liu, and Prasant Mohapatra. Distortion agnostic deep watermarking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13548–13557, 2020.
- [52] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. UDH: Universal deep hiding for steganography, watermarking, and light field messaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 10223–10234, 2020.

- [53] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. ReDMark: Framework for residual diffusion watermarking based on deep neural networks. *Expert Systems with Applications*, 146:113157, 2020.
- [54] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Ted Furon. The stable signature: Rooting watermarks in latent diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [55] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. volume 14, pages 1181–1193, 2019.
- [56] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. volume 23, pages 708–712, 2016.
- [57] Mehdi Yedroudj, Frederic Comby, and Marc Chaumont. Yedroudj-net: An efficient cnn for spatial steganalysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2092–2096, 2018.
- [58] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.