Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
AMO University of Bouira
Faculty of Exact Sciences
Department of Computer Science

Master's Thesis

in Computer Science

Specialization : *Information Systems and Software Engineering (ISIL)*

Theme

Robust Steganography Technique that can withstand data tampering or degradation

Supervised by

— Dr. Benaissi Sellami

Performed by

– Abid Akram

– Tahraoui Mustapha

Academic Year: 2025 – 2026

عرفان و إهداءٌ

الحمدُ لله... كلمةٌ تضيقُ عنها اللغاتُ كلُّها، وتعجزُ الحروفُ أن تُوفيَها حقَّها، هو الذي لمّا ضاقت بي الدروبُ وسَّعها، ولمّا أثقلني الحملُ خفَّفه، ولمّا كادت الأمنيةُ أن تذوي في داخلي نفخَ فيها من روح التوفيق فعاشت. **اللهم لك الحمد** على ما كان وما سيكون، لك الحمد على كلِّ سهرةٍ لم أُضِئ فيها سوى نورَ الأمل، وعلى كلِّ لحظةِ شكٍّ عبرتُها بيدِك لا بقوَّتي.

ثم أُقرُّ بفضل تلك الروح التي أبت الاستسلام، **نفسي** التي آمنت حين لم يؤمن بها أحد، وصمدت حين كان الصمود أصعبَ من الرحيل، وظلَّت تُهمهم في أعمق اللحظات سواداً: «أنت تستحق أن تصل.» وها قد وصلت.

وإلى من لم يكن نجاحي يوماً خبراً عادياً في عيونهم، إلى من دعَوا لي في السرِّ أكثر مما هنَّأوني في العلن، **إلى والديَّ الكريمَين** --- لستُ أُهديكما هذا النجاح، بل أُعيدُه إليكما، فهو منكما بدأ، وإليكما يعود.

وإلى من صنعوا من الأيام الثقيلة شيئاً يُطاق، **أصدقائي** وكلِّ من مدَّ يده دون أن أطلب --- أنتم لستم هامشاً في هذه القصة، **أنتم جزءٌ من حبرها.**

هذه ليست نهاية رحلة، بل هي أوَّل سطرٍ في فصلٍ لم يُكتب بعد.

Abstract

In the digital age, the exponential growth of multimedia communication has made information security a critical concern. Steganography, the art of hiding secret messages within innocent-looking cover media, has emerged as a powerful tool for covert communication. However, existing steganographic methods remain highly vulnerable to unintentional distortions and deliberate attacks, raising a fundamental question: *how can hidden data survive real-world degradations while remaining imperceptible?*

This thesis proposes **DASRS** (Damage Aware Self-Recovering Steganography), a robust steganographic framework that prioritizes payload survivability over mere concealment. The message is fragmented into six data chunks plus one XOR parity chunk, with each fragment protected by Reed–Solomon error correction and embedded simultaneously across four independent transform domains distributed across the YCbCr color space. A damage-aware cascade extraction with triple-validation and automatic XOR reconstruction enables partial message recovery even under severe compound distortions.

Experimental evaluation against signal processing, geometric, and compound attacks demonstrates that DASRS achieves substantially higher robustness than classical single-domain baselines while maintaining acceptable imperceptibility. The framework offers a principled, modular alternative to existing approaches through structural redundancy rather than attack-specific optimization.

Keywords: robust steganography, multi-domain embedding, error correction, cascade extraction, payload survivability

ملخص

في العصر الرقمي، أدى النمو المتسارع في وسائل التواصل والاتصال المرئي إلى جعل أمن المعلومات من أبرز التحديات التقنية الراهنة. وقد برز علم إخفاء المعلومات، الذي يقوم على تضمين رسائل سرية داخل وسائط رقمية عادية، بوصفه أداةً فعّالة للتواصل الآمن والسري. غير أن الأساليب القائمة تبقى هشّة أمام التشويهات غير المقصودة والهجمات المتعمدة، مما يطرح تساؤلاً جوهرياً: كيف يمكن للبيانات المخفية أن تصمد في مواجهة التدهور الحقيقي مع الحفاظ على سريتها؟

تقترح هذه الأطروحة نظام **DASRS** (إخفاء المعلومات ذاتي الاسترداد المدرك للأضرار)، وهو إطار عمل متين يضع قابلية البقاء للبيانات المضمّنة فوق مجرد الإخفاء. يُجزَّأ المحتوى السري إلى ستة أجزاء بيانات وجزء تحقق إضافي باستخدام XOR، ويُحمى كل جزء بتصحيح أخطاء Reed-Solomon، ثم يُضمَّن في آنٍ واحد عبر أربعة مجالات تحويل مستقلة موزعة على فضاء اللون YCbCr. كما يتيح الاستخراج التسلسلي المدرك للأضرار مع التحقق الثلاثي وإعادة البناء التلقائي عبر XOR استرداداً جزئياً للرسالة حتى في ظل تشويهات مركّبة حادة.

أثبت التقييم التجريبي في مواجهة هجمات معالجة الإشارة والهجمات الهندسية والمركّبة أن DASRS يحقق متانةً أعلى بكثير مقارنةً بالأساليب الكلاسيكية أحادية المجال، مع الحفاظ على مستوى مقبول من عدم الإدراك البصري. ويُقدّم الإطار بديلاً منهجياً ومرناً للمقاربات الموجودة، مستنداً إلى التكرار البنيوي عوضاً عن التحسين الموجّه نحو هجوم بعينه.

الكلمات المفتاحية: إخفاء المعلومات المتين، التضمين متعدد المجالات، تصحيح الأخطاء، الاستخراج التسلسلي، قابلية بقاء البيانات المضمّنة.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
DASRS	Damage Aware Self-Recovering Steganography
PRR	Payload Recovery Rate
RS	Reed–Solomon
CRC	Cyclic Redundancy Check
XOR	Exclusive OR
QIM	Quantisation Index Modulation
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
DFT	Discrete Fourier Transform
IDCT	Inverse Discrete Cosine Transform
IDWT	Inverse Discrete Wavelet Transform
2D-DCT-II	Two-Dimensional DCT Type II
JPEG	Joint Photographic Experts Group
WebP	Web Picture format
PNG	Portable Network Graphics
BGR	Blue-Green-Red (colour space)
RGB	Red-Green-Blue (colour space)
YCbCr	Luma-Chroma Red-Chroma Blue (colour space)
HVS	Human Visual System
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
MSE	Mean Squared Error
NC	Normalised Correlation
BER	Bit Error Rate
bpp	bits per pixel
QF	Quality Factor
LL	Low-Low (approximation sub-band)
LH	Low-High (horizontal detail sub-band)
HL	High-Low (vertical detail sub-band)
HH	High-High (diagonal detail sub-band)
DC	Direct Current (zero-frequency coefficient)
AC	Alternating Current (non-zero-frequency coefficients)
AWGN	Additive White Gaussian Noise
S&P	Salt and Pepper (noise)
EQ	Equalisation
ECC	Error Correction Code (also Error-Correcting Code)
LDPC	Low-Density Parity-Check
BCH	Bose–Chaudhuri–Hocquenghem (codes)
RAID	Redundant Array of Independent Disks
GAN	Generative Adversarial Network

(Continued from previous page)

Abbreviation	**Definition**
CNN	Convolutional Neural Network
INN	Invertible Neural Network
LDM	Latent Diffusion Model
MBRS	Mini-Batch of Real and Simulated JPEG Compression
LIDS	Low-frequency Image Deep Steganography
PRIS	Practical Robust Invertible Steganography
SRNet	Steganalysis Residual Network
SIFT	Scale-Invariant Feature Transform
PRNG	Pseudo-Random Number Generator
UTF-8	8-bit Unicode Transformation Format
GF(2^8)	Galois Field of 2^8 elements
SNR	Signal-to-Noise Ratio
ISO	International Organization for Standardization
IEEE	Institute of Electrical and Electronics Engineers

Introduction

Unlike cryptographic systems, which render message content unintelligible through mathematical transformation while openly acknowledging that a protected message exists, steganographic systems make the act of communication itself imperceptible. Digital watermarking, though it employs similar embedding mechanisms, pursues a distinct objective: asserting and verifying intellectual ownership rather than establishing a covert channel. 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 constitutes a discipline of covert data transmission in which a secret message is embedded within an ordinary digital carrier, leaving no observable indication that any communication has taken place. Etymologically, the word steganography combines two ancient Greek roots: steganos, denoting concealment or protection, and graphein, meaning to write — together conveying the notion of writing that hides itself[4]. In the context of image steganography, secret information is embedded exclusively within digital images, which serve as the cover medium for hidden communication.

0.1 Problem Statement

From the introduction 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.

0.2 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 [5] and Bit Error Rate (BER) [6], under realistic attack scenarios.

0.3 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.

Thesis Structure

This thesis is organized into five chapters, each building upon the preceding material to present the complete DASRS framework, its implementation, and evaluation. The structure is as follows:

Chapter 1: Literature Review. Establishes the theoretical foundations of image steganography, covering spatial and transform domain techniques (DCT, DWT, DFT), robustness-oriented approaches including error correction coding and synchronization mechanisms, and recent advances in deep learning-based steganography (2018–2025). The chapter concludes with a unified threat model and evaluation criteria that motivate the need for a payload-survivability-oriented framework.

Chapter 2: Proposed Framework and Implementation. Presents the complete DASRS architecture organized around three pillars: (i) structured payload design with six data fragments, one XOR parity fragment, per-fragment Reed–Solomon coding, and CRC-16 integrity verification; (ii) multi-domain redundant embedding across four independent domains (Spatial QIM on Y, DCT comparison on Y, DCT comparison on Cr, Haar DWT Level-2 QIM on Cb) with zone partitioning and embedding strength calibration; and (iii) damage-aware cascade extraction with triple validation and automatic XOR reconstruction. The chapter introduces the Payload Recovery Rate (PRR) as a novel metric for quantifying partial recovery.

Chapter 3: Implementation and Experimental Setup. Documents the Python prototype implementation, the development environment (Debian GNU/Linux, Jupyter Notebook, six open-source libraries), the two single-domain baselines (DCT-only and DWT-only with per-message Reed–Solomon), the USC-SIPI cover image dataset (eleven 512×512 images), the 49-byte test payload, and the complete 43-attack battery (33 single attacks + 10 compound attacks) with reproducible parameters.

Chapter 4: Results and Discussion. Reports the experimental evaluation across all three systems: embedding imperceptibility (PSNR, SSIM, histogram correlation), single-attack robustness (28/33 full recovery for DASRS), compound-attack robustness (8/10 full recovery), per-category analysis, Bit Error Rate comparison against twelve state-of-the-art methods (2018–2025), and domain cascade usage analysis. The chapter validates the three core design principles and discusses current limitations.

Chapter 1

Literature Review

1.1 Fundamentals of Image Steganography

Image steganography refers to techniques that embed secret information within digital images while preserving the visual appearance of the carrier. The field has evolved along three principal paradigms: classical methods that operate in the spatial or transform domain with hand-engineered rules; deep learning methods that replace these rules with end-to-end trainable neural networks; and hybrid methods that combine classical embedding mechanisms with deep learning components for adaptive region selection or post-processing refinement.

1.1.1 Imperceptibility, Capacity, and Robustness Trade-off

Any practical steganographic system must simultaneously satisfy three competing demands — visual transparency, embedding capacity, and resistance to degradation — that pull against one another. Strengthening any single dimension consumes resources or tolerances that the other two rely upon, making the mutual balance a fundamental design constraint rather than an engineering shortcoming. This pattern is often represented in a triangular relationship and is one of the main challenges in designing a steganographic system [7].

- **Imperceptibility** refers to the degree to which a stego-image remains visually and statistically indistinguishable from the corresponding cover image. Methods which aggressively embed data could introduce artifacts that affect either the statistical distribution or signal quality [8].
- **Payload capacity** denotes the amount of secret information that can be embedded within a cover image. High-capacity embedding schemes often sacrifice robustness in order to maximize payload [9].
- **Robustness** describes the ability of the embedded information to survive image degradation caused by signal processing operations or intentional attacks. Achieving robustness usually requires redundancy, structured embedding, or the use of stable image components, which in turn reduces the effective payload capacity [10].

In the context of robust steganography, robustness is frequently prioritized over capacity, particularly in applications where data recovery is more critical than transmission efficiency. Balancing these three competing objectives remains a key research challenge and motivates the exploration of resilience-oriented steganographic frameworks.

1.2 Spatial Domain Techniques

Spatial domain approaches embed the secret by directly modifying the intensities of pixels of the cover image. LSB substitution and its variants are the most studied families due to their simplicity, low computational complexity, and high embedding capacity [9]. In LSB-based methods, the

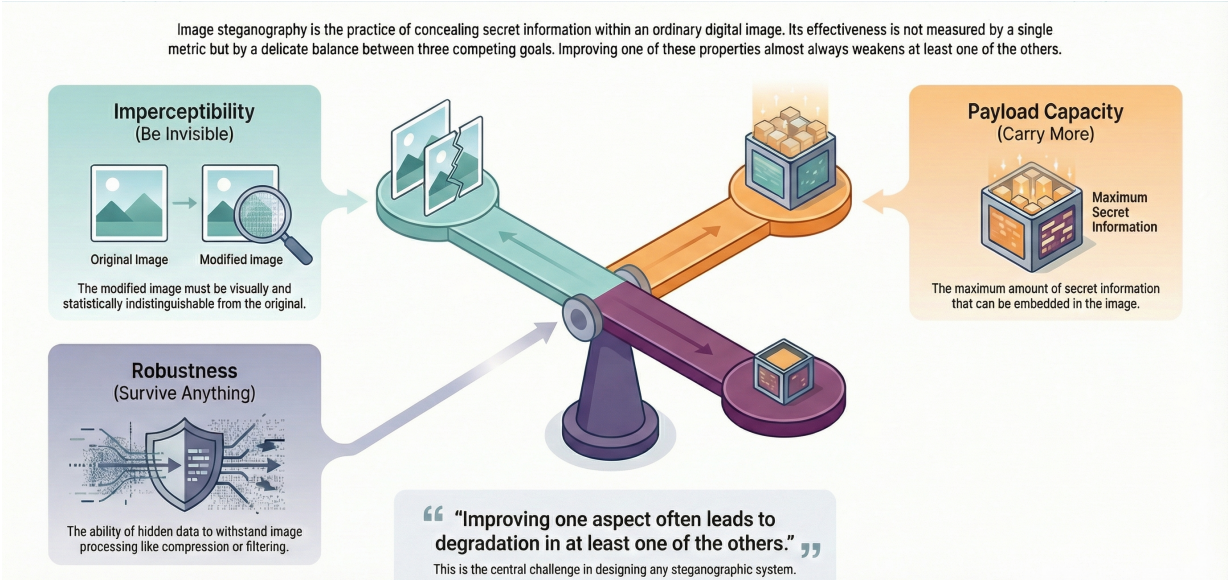


Figure 1.1: Steganography Trilemma: Three Competing Goals

least significant bits of pixel values are changed to convey secret information, leading to minimal perceptible distortion.

Despite their advantages, spatial domain techniques suffer from serious robustness drawbacks. Since the embedding procedure acts directly on pixel values, the hidden data is highly susceptible to common image processing manipulations. Even light lossy compression or noise addition can cause partial or complete loss of the embedded message [8]. Hence, most spatial domain methods are unsuitable for applications requiring robustness against image degradation.

1.3 Transform Domain Techniques

Transform domain techniques embed secret information within the coefficients produced by a mathematical transformation of the image. The core advantage lies in the properties of the human visual system, which exhibits varying sensitivity to different frequency components: modifications to certain frequency bands are far less perceptible than equivalent changes in the spatial domain [10].

The three most widely adopted transforms in image steganography are the Discrete Cosine Transform (DCT), the Discrete Wavelet Transform (DWT), and the Discrete Fourier Transform (DFT). Each transform decomposes the image into a different frequency representation, offering distinct trade-offs between robustness, imperceptibility, and resistance to specific attack types.

1.3.1 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. DCT-based steganography embeds secret information within the frequency coefficients of an image, exploiting the characteristics of the human visual system [7, 10].

The DCT converts spatial image data into a frequency-domain representation by dividing a grayscale or luminance image into non-overlapping 8×8 pixel blocks and transforming each independently. The resulting coefficients are categorized as low-frequency (average intensity and coarse structures), mid-frequency (edges and moderate texture), and high-frequency (fine details and noise-like components).

For color images, the RGB image is converted to YCbCr and embedding is performed in the luminance (Y) channel. Each 8×8 block is transformed using the 2D-DCT-II:

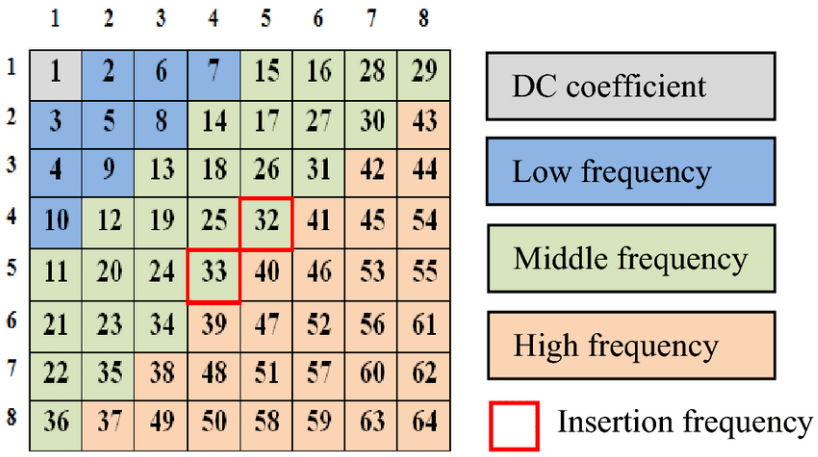


Figure 1.2: DCT coefficient distribution showing low-, mid-, and high-frequency components in an 8×8 block

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{i=0}^{7}\sum_{j=0}^{7} f(i,j)\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right] \tag{1.1}$$

where $C(k) = 1/\sqrt{2}$ if $k = 0$, else 1. Secret bits are embedded by modifying selected **mid-frequency coefficients** — high-frequency coefficients are discarded during compression while low-frequency/DC coefficients are too sensitive to changes. After embedding, the inverse DCT reconstructs the stego-image. Extraction mirrors the embedding steps: convert to YCbCr, apply DCT, locate the same mid-frequency coefficients, and recover the hidden bits.

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

Figure 1.3: DC and AC coefficient on DCT Transformation

DCT Method Benefits

- Robustness against JPEG compression is significantly improved.
- Embedding in the frequency domain allows better imperceptibility than spatial-domain methods.
- Compatibility with widely used image compression standards.

Limitations and Challenges

- Limited robustness against geometric distortions such as cropping, resizing, and rotation.
- Sensitivity to block misalignment caused by image re-sampling.
- Significant trade-off between payload capacity and robustness.

Although DCT techniques offer substantially increased robustness compared to spatial-domain approaches, DCT alone cannot guarantee successful recovery under highly adversarial conditions [3].

1.3.2 Discrete Wavelet Transform (DWT)-Based Steganography

The Discrete Wavelet Transform (DWT) decomposes the entire image through a hierarchical filter bank, producing sub-bands that jointly characterize spatial location and frequency content. This multi-resolution property makes DWT-based steganography more perceptually adaptable and more resilient to localized distortions than block-based approaches [11, 12].

Applying the DWT separably to image rows and columns yields four sub-bands at each decomposition level j: **LL** (approximation, majority of image energy), **LH** (horizontal edges), **HL** (vertical edges), and **HH** (diagonal details). The LL sub-band can be recursively decomposed for a hierarchical multi-resolution representation.

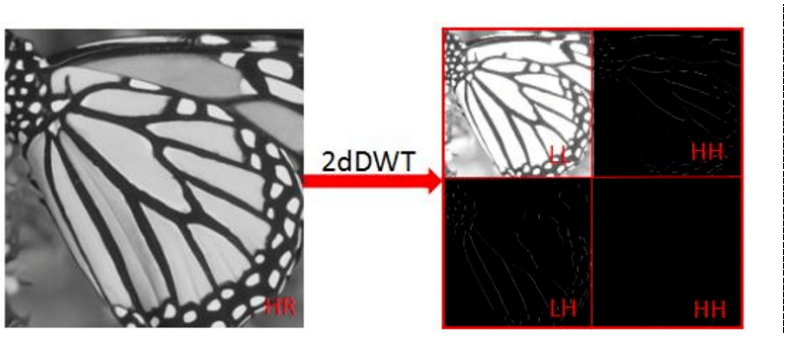


Figure 1.4: One-level 2D-DWT decomposition producing the LL, LH, HL, and HH sub-bands.

Embedding is performed in the mid-frequency detail sub-bands (typically LH and HL at level 1 or 2). The LL sub-band is avoided as modifications cause strong perceptual distortion, and the HH sub-band is susceptible to compression-induced zeroing. A secret bit b is embedded using a quantization-based rule:

$$W'(m,n) = \begin{cases} \left\lfloor \dfrac{W(m,n)}{\Delta} \right\rfloor \cdot \Delta + \dfrac{\Delta}{4} & \text{if } b = 0 \\ \left\lfloor \dfrac{W(m,n)}{\Delta} \right\rfloor \cdot \Delta + \dfrac{3\Delta}{4} & \text{if } b = 1 \end{cases} \tag{1.2}$$

where Δ is the quantization step size. After embedding, the IDWT reconstructs the stego-image. Hidden bits are extracted by applying the same decomposition and recovering each bit via the inverse quantization rule: $\hat{b} = \lfloor W'(m,n)/\Delta \rfloor \bmod 2$.

Advantages of DWT-Based Steganography

- The wavelet decomposition's graduated spatial-frequency structure mirrors the way the human visual cortex processes imagery at different scales simultaneously.
- Greater robustness against localized distortions and partial image cropping.
- Superior spatial-frequency localization compared to block-based transforms.

Limitations and Challenges

- Susceptibility to geometric transformations such as resizing and rotation.
- Higher computational complexity than spatial-domain methods.
- Vulnerability to aggressive multi-stage compression attacks.

While DWT techniques offer more robust and perceptually adaptive solutions than spatial-domain and block-DCT methods, additional resilience mechanisms are required for reliable data recovery under extreme image degradation [10].

1.3.3 Discrete Fourier Transform (DFT)-Based Steganography

Applying the DFT to an image yields a single global frequency map in which every spectral coefficient reflects the contribution of every pixel. This stands in contrast to the DCT, which partitions the image into independent blocks before transformation, and the DWT, which produces a layered multi-scale decomposition rather than a unified spectrum [13, 11]. This global nature confers partial invariance to geometric transformations such as rotation and translation, but means that a single spatial modification affects every frequency coefficient in the spectrum [14, 3].

The 2D DFT transforms a spatial-domain image $I(x, y)$ into a complex-valued frequency matrix:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cdot e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{1.3}$$

Each coefficient separates into magnitude $|F(u, v)|$ and phase $\phi(u, v)$. Research shows that the phase spectrum preserves the majority of perceptual information, so steganographic modifications are applied to the **magnitude spectrum only** [15].

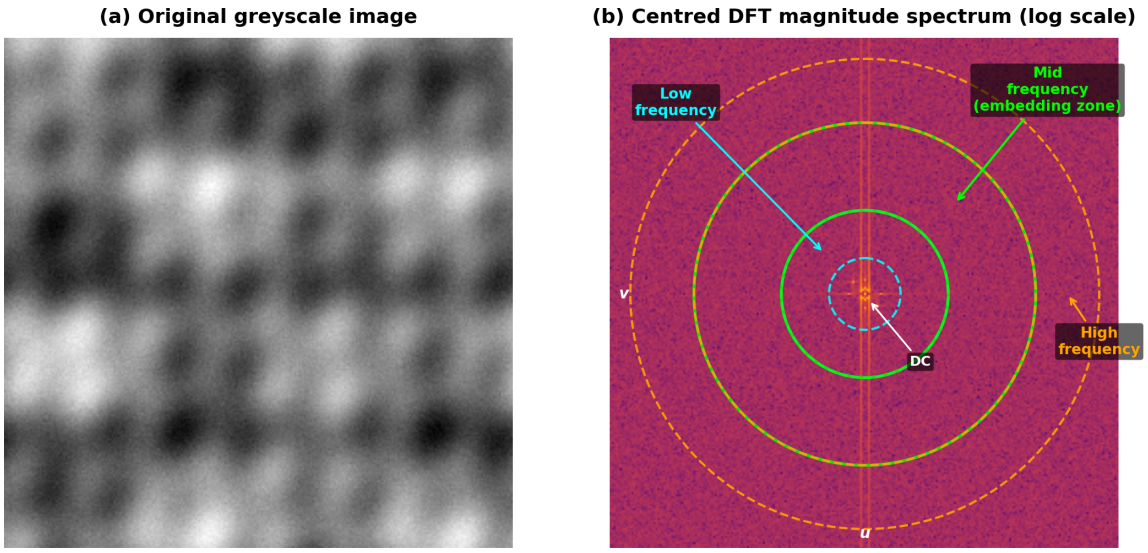


Figure 1.5: Centred 2D DFT magnitude spectrum of a natural image (log scale). Steganographic embedding targets the mid-frequency ring to balance robustness and imperceptibility.

Embedding selects mid-frequency coefficients defined by a radial band $\mathcal{R} = \{(u, v) : r_{\min} \leq \sqrt{u^2 + v^2} \leq r_{\max}\}$, providing partial robustness to rotation. Secret bits modify the magnitude via $|F'(u, v)| = |F(u, v)| + \alpha \cdot b \cdot |F(u, v)|$, while preserving the phase. The conjugate symmetry constraint $F(M - u, N - v) = F^*(u, v)$ must be maintained to ensure a real-valued inverse DFT output.

Advantages of DFT-Based Steganography

- **Translation invariance:** Spatial shifts do not affect the magnitude spectrum [13].
- **Rotation robustness:** Embedding in circularly symmetric spectral regions provides robustness to image rotation [14, 16].
- **No blocking artefacts:** Processing the whole image avoids tile-boundary artefacts.
- **Spread-spectrum compatibility:** DFT coefficients serve directly as carriers for pseudo-random spreading sequences [17].

Limitations and Challenges

- **Cropping vulnerability:** Cropping destroys the global periodicity assumption, affecting all frequency coefficients [3].
- **JPEG sensitivity:** JPEG recompression alters DFT coefficients in a complex, block-dependent manner [10].
- **Conjugate symmetry constraint:** Effectively halves the number of independently usable embedding positions.
- **Lower capacity than DCT:** Only a single global spectrum is available versus per-block coefficients [7].

DFT-based steganography is best suited to applications where rotation and translation robustness are primary requirements [14, 10].

1.4 Robustness-Oriented Approaches

Achieving robustness in image steganography requires 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 reviews the principal strategies developed toward this goal.

1.4.1 Error Correction Coding for Steganographic Robustness

A widely adopted strategy for improving robustness 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, ECC allows the receiver to detect and correct a bounded number of corrupted bits [10].

Block codes. Unlike binary codes that treat each bit independently, Reed–Solomon codes group bits into larger symbols and reason about corruption at the symbol level. A codeword of length n symbols carries k information symbols and n−k redundancy symbols, conferring the ability to locate and restore any group of up to $\lfloor (n-k)/2 \rfloor$ damaged symbols — a property that makes them naturally resilient against the concentrated block-level damage introduced by lossy compression[3].

Convolutional and turbo codes. Convolutional encoding extends each output bit's dependence across a sliding window of prior inputs, creating a trellis structure that the Viterbi algorithm exploits to achieve maximum-likelihood sequence decoding in linear time [6]. Turbo codes approach the Shannon capacity limit and have been applied in watermarking systems demanding near-optimal error recovery [17].

LDPC codes. LDPC codes are defined by a sparse parity-check matrix whose structure enables iterative message-passing decoding to converge near the Shannon limit, achieving this performance at a computational cost that scales more favourably than turbo decoding when codewords are long [9].

Limitations of ECC alone. Despite their effectiveness, 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, producing burst erasures at unknown locations [10]. Under these conditions, even maximum-distance-separable codes fail. This motivates the complementary mechanisms discussed below.

1.4.2 The Synchronization Problem

Among the obstacles that robust steganographic systems must overcome, the failure of spatial correspondence between embedding and extraction — commonly termed desynchronization — carries the severest practical consequences, as it can render even perfectly preserved embedded bits unrecoverable [14]. A steganographic system is synchronized when the extractor can reliably locate the embedding positions within the received image. Geometric transformations invalidate fixed coordinate systems, causing qualitatively different failure from ordinary bit corruption.

Causes and Manifestations

- **Cropping** permanently destroys embedded fragments and prevents reconstruction of the coordinate grid [3].
- **Resizing and resampling** shift pixel positions and distort block boundaries; even a one-pixel shift causes catastrophic BER increase in DCT-based schemes [10].
- **Rotation** introduces a global angular offset that prevents coefficient identification unless the embedding domain is rotation-invariant [14].
- **Compound attacks** combine multiple operations sequentially. Social-media platforms typically resize, apply JPEG compression, and sometimes crop in a single automated pipeline [7].

Template-Based Synchronization

A synchronization approach that co-embeds a structured pilot signal with the payload enables the receiver to measure how the image has been geometrically deformed and to apply the corresponding inverse mapping before attempting bit recovery [14]. Kutter [18] demonstrated that circularly symmetric patterns in the Fourier domain are particularly robust for template-based registration. Solachidis and Pitas [19] proposed ring-shaped patterns in the 2D DFT magnitude spectrum, exploiting the Fourier shift theorem for translation invariance.

Feature-Based Alignment

An alternative approach anchors embedding positions to stable image features — corners, edges, or salient regions — that can be reliably re-detected after geometric distortion [14]. Feature detectors such as SIFT have been adopted to compute a canonical coordinate frame from image content itself [3, 10]. Severe cropping or blurring may destroy the features used for alignment.

1.4.3 Robust Steganography vs. Digital Watermarking

Robust steganography and digital watermarking share a common toolbox but differ fundamentally in objectives, threat models, and success criteria [10, 20].

Digital watermarking is designed for *ownership verification*: the embedded mark is expected to be detectable by an authorized verifier, and its persistence is the primary success criterion [3]. Watermarking schemes can afford stronger, more visible embedding and are evaluated on detectability under attack rather than undetectability.

Steganography requires *covert communication*: the existence of the hidden message must remain undetectable to unauthorized observers [8, 9]. Techniques that enhance robustness — redundancy, high embedding strength, structured payload distribution — tend to introduce statistical artifacts that increase steganalytic detection risk [7]. The steganographer must simultaneously balance imperceptibility, robustness, and undetectability, whereas a watermarker need only balance the first two.

Robust steganography occupies the intersection of these paradigms, and this dual constraint motivates the payload-oriented framework proposed in Chapter 2.

1.4.4 Steganalysis and the Detectability–Robustness Trade-off

A complete treatment of robustness must acknowledge the threat of steganalysis because robustness-enhancing design choices often increase detectability [7, 3].

Classical steganalysis. Chi-squared analysis [8] detects the pairing of adjacent pixel values caused by LSB substitution. RS analysis examines the ratio of regular and singular pixel groups [8]. These attacks are algorithm-specific and fail against others.

Universal feature-based steganalysis. Modern steganalysis extracts a high-dimensional feature vector and trains a machine learning classifier to distinguish cover from stego-images without requiring knowledge of the specific embedding algorithm [7]. The Rich Model of Fridrich and Kodovský [7] computes co-occurrence statistics of pixel prediction errors, producing feature vectors with tens of thousands of dimensions. Paired with ensemble classifiers, these feature sets achieve near-perfect detection even at very low embedding rates.

Implications for robust steganographic design. The detectability–robustness trade-off imposes a practical ceiling on embedding strength. Stronger embedding improves survivability under attack but leaves larger statistical footprints. This motivates design approaches that achieve robustness through *structure* — fragmentation, cross-fragment relationships, and domain diversity — rather than through raw embedding strength.

1.4.5 Deep Learning–Based Robust Steganography

Deep learning approaches to steganography — including encoder–decoder architectures, generative adversarial frameworks, and hybrid transform-network methods — have emerged as a significant research direction since 2018. A comprehensive review of these methods and the reasons this thesis adopts a classical framework instead is provided in Section 1.8.

1.4.6 Limitations of Current Approaches

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 tolerate the *compound*

of geometric desynchronization, partial data erasure, and signal-level noise that characterizes real-world environments.

Table 1.1 summarizes the strengths and limitations of each major category.

Table 1.1: 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 1.1 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 motivates the self-recovering framework proposed in Chapter 2.

1.5 Evaluation Criteria in Steganographic Systems

The design and evaluation of any steganographic system requires quantitative metrics measuring performance along three fundamental axes: imperceptibility, robustness, and steganalysis resistance. This section provides a comprehensive treatment of the evaluation criteria used throughout this thesis.

1.5.1 Imperceptibility Metrics

Mean Squared Error (MSE)

The MSE is the fundamental measure of pixel-level distortion between cover image C and stego-image S of dimensions $M \times N$:

$$\mathrm{MSE}(C,\, S) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \Big[C(i,j) - S(i,j)\Big]^2 \tag{1.4}$$

For colour images, MSE is computed channel-wise and averaged. While a lower MSE indicates less distortion, equal MSE values can correspond to perceptually very different distortion types, making it an insufficient sole measure.

Peak Signal-to-Noise Ratio (PSNR)

PSNR has achieved near-universal adoption as the benchmark measure of embedding transparency in the steganographic literature, owing to its straightforward interpretation and its direct relationship to mean squared pixel-level distortion [5]:

$$\mathrm{PSNR}(C,\, S) = 10 \cdot \log_{10}\!\left(\frac{L^2}{\mathrm{MSE}(C,\, S)}\right) \quad [\mathrm{dB}] \tag{1.5}$$

where $L = 255$ for 8-bit images. In the steganographic literature, a stego-image with PSNR > 30 dB is broadly regarded as exhibiting good visual quality [21], with values in the range 30–35 dB considered acceptable and those above 40 dB excellent; PSNR treats all pixel positions and spatial frequencies as equally important, whereas the human visual system (HVS) is significantly more sensitive to distortion in homogeneous regions than in textured areas [22].

Structural Similarity Index Measure (SSIM)

SSIM quantifies perceptual similarity by measuring how well the stego-image preserves the luminance, contrast, and spatial structure of the original — properties to which human vision is known to be particularly sensitive — rather than treating all pixel differences as equally distorting [22]. Computed locally over 11×11 windows:

$$\mathrm{SSIM}(\mathbf{x},\, \mathbf{y}) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \cdot \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \cdot \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{1.6}$$

SSIM ranges from -1 to 1; values > 0.99 indicate excellent imperceptibility, 0.97–0.99 are good, 0.90–0.97 are acceptable. SSIM is generally a more reliable predictor of perceived quality than PSNR for compression-type distortions. Both metrics are reported in the experimental evaluation of Chapter 4 as they capture complementary aspects: PSNR reflects energy distortion while SSIM captures structural degradation.

1.5.2 Robustness Metrics

Bit Error Rate (BER)

The BER measures the fraction of payload bits incorrectly recovered. For original message $M = [m_1, \ldots, m_K]$ and extracted message $\hat{M}$:

$$\mathrm{BER}(M,\, \hat{M}) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\big[\hat{m}_k \neq m_k\big] \tag{1.7}$$

BER $= 0$ indicates perfect recovery; BER ≤ 0.05 indicates good robustness; BER ≈ 0.5 indicates complete failure (random output). When protected by an ECC with correction capacity t over codewords of length n, perfect message recovery is possible if and only if BER $< t/n$.

Normalised Correlation (NC)

The NC provides a continuous measure of recovery fidelity robust to systematic bit inversions [10]. After bipolar encoding ($0 \to -1$, $1 \to +1$):

$$\mathrm{NC}(M,\, \hat{M}) = \frac{1}{K} \sum_{k=1}^{K} \bar{m}_k \cdot \hat{\bar{m}}_k \tag{1.8}$$

NC $= 1.0$ indicates perfect recovery; NC $= 0.0$ indicates complete failure; NC $= -1.0$ indicates systematic polarity reversal (recoverable by inversion, but misreported as BER $= 1$). The relationship NC $= 1 - 2 \cdot$ BER holds for independent errors. Both metrics are reported to facilitate comparison with the watermarking literature.

1.5.3 Unified View of Evaluation Criteria

Table 1.2 summarises all evaluation metrics, their definitions, and performance thresholds adopted in this thesis.

Table 1.2: Summary of evaluation metrics. 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. (1.6)	$[-1, 1]$	> 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. (2.19)	[0, 1]	> 0.7 (proposed)
bpp	Capacity	bits embedded per pixel	(0, 8]	Application-dependent

These metrics, taken together, provide a comprehensive multi-dimensional assessment of steganographic performance. In Chapter 4, each metric is reported for the proposed framework and for classical baselines under each attack scenario, enabling rigorous comparative evaluation that respects the multi-objective nature of the steganographic design problem.

1.6 Attacks and Image Degradation Techniques

In real-world deployment, stego-images routinely pass through social media platforms, messaging applications, and format conversion pipelines, each applying processing operations that may destroy or degrade the embedded payload. A robust steganographic system must be designed and evaluated against a formally specified threat model [3, 10].

1.6.1 Signal Processing Attacks

Signal processing attacks modify pixel intensity values while preserving the geometric structure of the image. The appropriate countermeasure is error correction coding, which can recover a payload from a moderate fraction of corrupted bits provided synchronisation is maintained.

Lossy Compression: JPEG

Of all signal-level distortions an embedded payload may encounter, JPEG recompression poses the greatest practical threat: it operates transparently in the background of nearly every image sharing platform, content management system, and consumer camera pipeline, making avoidance effectively impossible in real-world deployment [23]. It operates in the DCT domain and introduces distortion through coefficient quantisation:

$$\tilde{F}(u,v) = Q(u,v) \cdot \left\lfloor \frac{F(u,v)}{Q(u,v)} + 0.5 \right\rfloor \tag{1.9}$$

where $Q(u,v)$ is the quality-dependent quantisation step. Lower quality factors correspond to larger $Q(u,v)$ and greater information loss, with reconstruction error bounded by $|\Delta F(u,v)| \leq Q(u,v)/2$.

The impact on embedded signals depends critically on placement: high-frequency embeddings are nearly entirely overwritten (BER typically exceeds 40% at QF = 70 for spatial LSB methods); mid-frequency embeddings survive substantially better provided the embedding strength $\alpha > Q(u,v)/2$; low-frequency coefficients survive even aggressive compression but produce perceptible intensity shifts.

Additive Noise

Additive white Gaussian noise (AWGN) arises from sensor imperfections or deliberate noise injection [5]. The received signal is $\tilde{I}(x,y) = I(x,y) + \eta(x,y)$, where $\eta(x,y) \sim \mathcal{N}(0,\sigma_n^2)$. Unlike JPEG compression, AWGN distributes energy uniformly across all spatial frequencies. For spread-spectrum steganography, the decoding SNR is $\text{SNR}_{\text{dec}} = \alpha^2 N/\sigma_n^2$, showing that spreading gain N provides a factor-of-N robustness improvement [17].

Filtering Attacks

Low-pass filtering applies a convolution kernel to the image, attenuating high-frequency content [5]. In the frequency domain, convolution becomes multiplication: $\tilde{F}(u,v) = F(u,v) \cdot H(u,v)$, where $H(u,v) \in [0,1]$. A steganographic signal at frequency (u_0,v_0) survives only as fraction $H(u_0,v_0)$ of its original magnitude; for Gaussian blur with $\sigma = 2.0$, high-frequency components are reduced by over 90%. Median filtering is additionally destructive because isolated modified pixels are voted out entirely.

Colour Space Conversion and Format Conversion

Social media platforms frequently convert images between colour spaces or formats, redistributing energy between luminance and chrominance components and potentially disrupting channel-specific embeddings [10]. A full print-scan cycle introduces blur, noise, perspective distortion, and colour shift simultaneously [2].

1.6.2 Geometric Attacks

Geometric transformations act on the coordinate layout of the image rather than on pixel brightness, displacing or removing the spatial anchors that the extractor relies upon to locate where bits were written. The consequence is positional misalignment: the embedded data may be physically present in the image but is read from incorrect locations, producing extraction errors that are qualitatively different from ordinary bit corruption [14, 10].

Cropping

Cropping removes a contiguous rectangular region, permanently deleting all pixels and embedded data within it. For a system embedding N_F fragments uniformly distributed across the image, the expected fragments destroyed by a crop of fraction ρ_{crop} is $\mathbb{E}[\text{fragments lost}] = N_F \cdot \rho_{\text{crop}}$. Beyond direct data loss, cropping destroys the coordinate reference frame: block-based extractors (DCT, DWT) depend on knowing the exact image origin to reconstruct the block grid, so even a single-pixel crop offset produces a near-random BER, as illustrated in Figure 1.6.

Resizing and Resampling

Resizing changes the spatial resolution through pixel interpolation. For a scaling factor s, the resized image is $\tilde{I}(x,y) = \sum_{m,n} I(m,n)\, k(x/s - m,\, y/s - n)$, where $k(\cdot)$ is an interpolation kernel (nearest-neighbour, bilinear, or bicubic). Resampling has two effects: the interpolation kernel acts as a low-pass filter, and non-integer scale factors change the pixel count, destroying the one-to-one correspondence between embedding and extraction positions. For DCT-based embedding, this shifts the 8×8 block boundaries to non-integer positions [5].

Rotation

Rotation transforms the image by an angle θ about a centre point. Even a small rotation of $\theta = 1$ř displaces pixels near the image boundary by several pixels, making it impossible for a block-based

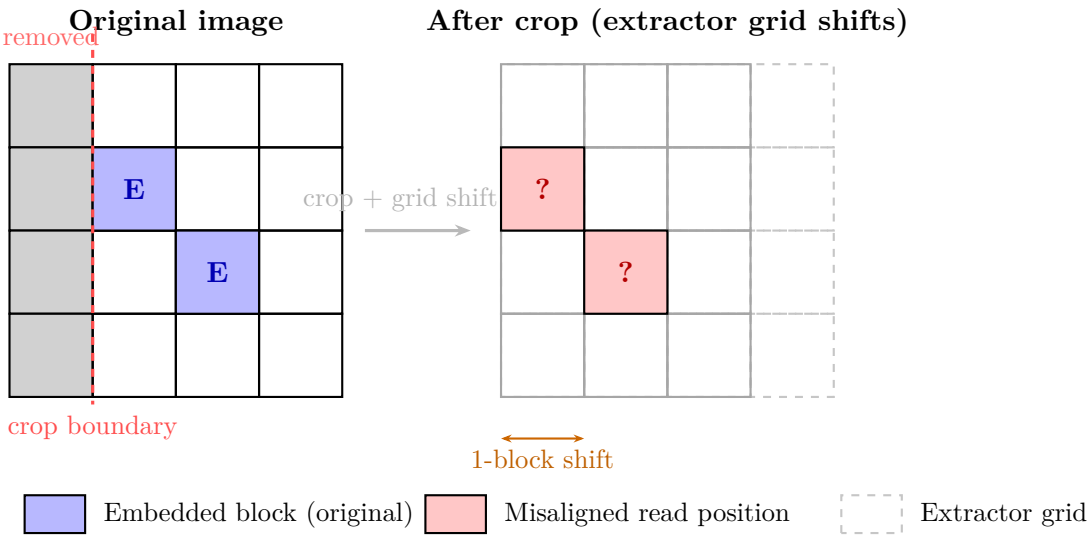


Figure 1.6: Effect of cropping on block-based DCT embedding. Column 0 is removed; the extractor grid shifts by one block, causing misaligned reads (marked ?) even though the embedded blocks E were not removed.

extractor to locate its grids. For DFT-based steganography, the Fourier magnitude spectrum rotates by the same angle, preserving energy at the same radial frequency — embeddings in circularly symmetric regions therefore survive rotation, while those at specific angular positions do not [13, 14].

Affine and Projective Transformations

Affine mappings (combinations of rotation, scaling, shearing, and translation) and projective transformations arise in print-scan scenarios. Both classes are highly destructive to all location-dependent steganographic methods and require either invariant-domain embedding or template-based re-synchronisation for recovery [10, 18].

1.6.3 Compound and Pipeline Attacks

In practice, stego-images are rarely subjected to a single isolated attack. Real-world distribution channels apply multiple operations sequentially, each building on previous damage [3].

Table 1.3: Documented image processing operations applied by major social media and messaging platforms. QF: JPEG quality factor. Compiled from [1, 2, 3].

Platform	JPEG recompr.	Resize	Crop	Format conv.	Approx. QF
WhatsApp	Yes	Yes (max 1600px)	Sometimes	JPEG	75–85
Instagram	Yes	Yes (max 1080px)	Yes (square)	JPEG	70–80
Twitter/X	Yes	Yes (max 4096px)	No	JPEG/WebP	85
Facebook	Yes	Yes	No	JPEG	70–85
Telegram	Yes	Sometimes	No	JPEG	90–95

A compound attack can be modelled as the sequential application of K degradation operators: $\tilde{I} = \mathcal{D}_K \circ \cdots \circ \mathcal{D}_1(I_s)$. Operators interact non-linearly — geometric distortion followed by JPEG compression is far more destructive than either alone, because the geometric operation shifts block boundaries and the subsequent JPEG operates on misaligned blocks. Studies have shown that a WhatsApp-equivalent pipeline reduces BER recovery of a standard DCT mid-frequency embedding from approximately 8% (JPEG alone) to over 35% [1, 10].

Figure 1.7: A typical social media compound attack pipeline. Each stage introduces a distinct type of degradation; the cumulative effect is far more destructive than any single attack alone.

1.6.4 Unified Attack Summary

Table 1.4 provides a comprehensive summary of all attack types, their degradation mechanisms, failure modes, and implications for the proposed framework.

The key asymmetry is that signal processing attacks introduce correctable bit-level errors (addressable by ECC), while geometric attacks destroy synchronisation (making ECC irrelevant). The proposed framework addresses both by combining structured spatial distribution of fragments with multi-domain embedding, maintaining a positive PRR under any compound attack within the system's erasure budget.

1.7 Threat Model

A rigorous evaluation of any steganographic system requires an explicit threat model that specifies the adversary's capabilities, the channel characteristics, and the conditions under which the system is expected to operate. This section formalises the threat model assumed throughout this thesis.

1.7.1 Adversary Model

The adversary is assumed to be *passive* in the sense of the prisoner's problem [24]: the adversary suspects that hidden communication may be occurring and attempts to detect or destroy the payload, but does not modify the stego-image in an adaptive, payload-aware manner. Specifically, the adversary:

- Knows that steganography may be in use, but does not know the specific algorithm, embedding domains, or zone assignment key;
- Can apply any combination of the signal processing, geometric, and compound attacks described in Section 1.6;
- Cannot access the metadata dictionary required for extraction;
- Does not possess the original cover image.

The adversary is *not* assumed to be active in the sense of an adaptive attacker who modifies the image specifically to target known embedding positions. This restriction is standard for robust steganography and distinguishes the threat model from that of adversarial watermarking, where the attacker may possess detailed knowledge of the embedding algorithm.

1.7.2 Channel Model

The communication channel is modelled as a sequence of potentially lossy and distorting operations applied automatically by digital image distribution platforms. The channel is:

- *Uncontrolled:* The sender has no influence over which operations are applied or in what order;
- *Non-adaptive:* The channel distortions are independent of the payload content;

Table 1.4: Unified summary of attack types and implications for the proposed self-recovering framework. BER: Bit Error Rate; ECC: Error Correction Code; PRR: Payload Recovery Rate.

Attack	**Degradation model**	**Failure mode**	**Survivable by**	**Framework implication**
JPEG compression	Coefficient quantisation (Eq. 1.9)	High BER in unstable coefficients	ECC, mid-freq. embedding	Embed in DCT mid-freq.; strength $> Q/2$
AWGN	Additive Gaussian	Random bit flips	ECC, SS spreading	ECC at fragment level corrects noise errors
Low-pass filtering	Convolution attenuation	Erasure of high-freq. embedding	Mid-freq. embedding	Avoid high-frequency subbands
Cropping	Spatial erasure + grid shift	Fragment loss + sync loss	Redundancy, fragmentation	Distributed fragments maximise PRR under crop
Resizing	Interpolation + scale shift	Block misalignment + signal attenuation	Invariant domains, multi-scale	Multi-scale redundant embedding reduces sensitivity
Rotation	Coordinate transform + interpolation	Global synchronisation loss	Log-polar DFT, templates	DFT magnitude used for geometric-invariant copies
Compound pipeline	Sequential composition	Cascaded signal + sync degradation	Structured redundancy only	Multi-domain distribution essential
Steganalysis	Statistical feature deviation	Channel exposure	Low distortion, structural embedding	Low per-coefficient distortion limits footprint

- *Compound-capable:* Multiple operations may be applied sequentially, with each stage operating on the output of the previous stage;
- *Synchronisation-disrupting:* Geometric operations may destroy the spatial alignment between embedding and extraction grids.

This channel model directly reflects the processing pipelines of social media platforms, messaging applications, and cloud storage services documented in Table 1.3.

1.7.3 Success Criteria

The steganographic system is required to satisfy three success criteria:

1. **Imperceptibility:** The stego-image must be visually and statistically indistinguishable from the cover image under normal viewing conditions. Quantitatively, PSNR $\geq$ 30 dB and SSIM ≥ 0.92 are required.
2. **Partial recoverability:** Under any attack within the evaluated battery, the system must return a non-empty message together with a PRR score quantifying the fraction recovered. Total failure (PRR = 0) is acceptable only for attacks that physically destroy the carrier (e.g., crop exceeding the erasure budget).
3. **Exact reconstruction:** When the XOR parity trigger conditions are satisfied (exactly one data fragment lost, parity fragment available), the missing fragment must be reconstructed with zero bit errors, not approximated or estimated.

The threat model explicitly excludes attacks that destroy the entire image or apply distortion beyond the evaluated battery. DASRS does not claim security against arbitrary adversarial manipulation; its robustness guarantee is conditional on the attack being representable within the compound-attack framework of Chapter 3.

1.8 Recent Advances in Deep Learning-Based Steganography

Since 2018, deep learning approaches have dramatically transformed the steganographic field. CNNs, GANs, and attention-based architectures promise simultaneous optimisation of imperceptibility, capacity, and robustness in ways classical methods cannot easily achieve. This section surveys the most influential systems published between 2018 and 2024, with attention to their robustness properties and limitations relative to the payload-survivability problem that motivates this work.

1.8.1 The Shift Toward End-to-End Learned Steganography

Classical steganographic systems are hand-engineered: every component is transparent and independently chosen. Deep learning replaces this modular pipeline with a single differentiable system trained end-to-end [25, 26]. An *encoder* network E_θ maps a cover image C and secret message M to a stego-image $S = E_\theta(C, M)$, while a *decoder* D_ϕ recovers the message $\hat{M} = D_\phi(\tilde{S})$ from the (possibly degraded) received signal. Training minimises:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{img}}(C, S) + \lambda_2 \mathcal{L}_{\text{msg}}(M, \hat{M}) + \lambda_3 \mathcal{L}_{\text{adv}} \tag{1.10}$$

The key advantage is that gradient back-propagation allows encoder and decoder to co-adapt through a differentiable noise model simulating real-world attacks during training.

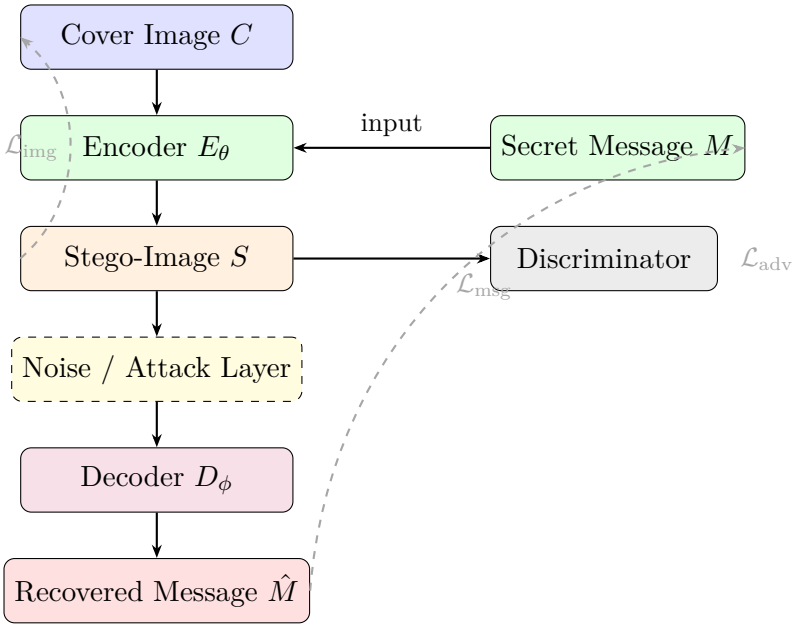


Figure 1.8: Generic encoder–decoder architecture for deep steganography. The noise layer simulates attacks during training, causing the encoder to learn attack-resistant embeddings.

1.8.2 HiDDeN: The Foundational Framework (2018)

The HiDDeN architecture by Zhu et al. established the modern template for learned steganography through three contributions that subsequent work has largely adopted as defaults: end-to-end convolutional networks that discover embedding strategies from training data rather than following hand-crafted rules; a differentiable surrogate for the transmission channel placed between encoder and decoder so that resistance to specific distortions emerges automatically through back-propagation; and an adversarial objective that penalises statistically detectable modifications.[25], including differentiable JPEG approximations, Gaussian noise, dropout, and Gaussian blurring. Third, a GAN-style discriminator loss encourages stego-images to be statistically indistinguishable from natural photographs [27]. HiDDeN achieves PSNR above 33 dB for 30-byte payloads under JPEG and Gaussian noise with BER below around $3-4\%$, significantly outperforming the classical F5 algorithm [28].

The key limitation is the distribution-shift problem: if the deployed system encounters attacks not represented in the noise layer (e.g. spatial warping or aggressive cropping), the encoder has learned no strategy to resist them.

1.8.3 SteganoGAN: High-Capacity GAN-Based Steganography (2019)

Zhang et al. [29] extended the HiDDeN framework with a focus on maximising capacity. The dense encoder achieves up to 4.4 bpp at PSNR 30.5 dB. However, SteganoGAN *does not include a noise layer*: training under ideal conditions produces visually excellent but highly fragile stego-images. Under JPEG QF $= 70$, BER rises above 40% — complete communication failure. This confirms that capacity-oriented deep models without explicit robustness training are unsuitable for uncontrolled channels, and illustrates a general principle: higher capacity invariably exploits fragile high-frequency regions discarded by lossy compression.

1.8.4 ReDMark: Robustness-First Deep Steganography (2021)

Das. [30] introduced ReDMark, the first deep steganographic system explicitly designed around robustness as the primary constraint. A U-Net encoder with skip connections anchors embedding to stable image features that compression preserves [31]. The training pipeline incorporates a differentiable JPEG approximation [32] and a spatial transformer network [33] simulating mild geometric

transformations. A dynamic loss-weight redistribution scheme adaptively adjusts λ_1 (image quality) versus λ_2 (message fidelity) based on current BER, preventing convergence to fragile embeddings. ReDMark achieves PSNR 36.2 dB at 30-bit payload with BER 1.4% after JPEG QF = 50.

Despite these improvements, ReDMark retains the distribution-shift vulnerability and provides no partial-recovery mechanism: if BER exceeds the decoder's capability, all output is unintelligible.

1.8.5 Recent Advances: 2021–2025

The period from 2021 to 2025 has witnessed three distinct research threads that push the boundaries established by HiDDeN, SteganoGAN, and ReDMark: (i) tighter JPEG robustness through training curriculum design, (ii) invertible and frequency-aware architectures that embed payload into compression-stable frequency regions, and (iii) a paradigm shift toward generative and diffusion-based steganography that replaces the cover-image editing model with cover-image synthesis.

JPEG-Specific Robustness Curricula

The fundamental difficulty of differentiating through JPEG compression — a piece-wise non-differentiable quantisation operation — motivated Jia et al. [1] to propose the *Mini-Batch of Real and Simulated JPEG Compression* (MBRS) framework. MBRS trains the encoder–decoder pair by randomly selecting, for each mini-batch, one of three noise-layer configurations: a differentiable JPEG simulator, an actual JPEG codec call, or a noise-free pass. This mixed-curriculum strategy forces the encoder to learn embeddings that survive both the gradients of the differentiable approximation *and* the true quantisation table applied by a real JPEG encoder. The resulting system achieves PSNR 38.1 dB at a 30-bit payload with BER 0.9% under JPEG QF = 70 — the best combined imperceptibility–robustness figure of its generation. Despite this, MBRS retains the distribution-shift vulnerability: attacks not represented in the three-configuration curriculum (e.g. spatial warping, aggressive cropping) degrade BER substantially, and, like all encoder–decoder systems surveyed, it provides no partial-recovery mechanism for partially destroyed stego-images.

Frequency-Domain Embedding: LIDS

A parallel line of work targets the high-frequency artifact problem identified in SteganoGAN. Chen et al. [34] observe that CNN-generated stego-images concentrate embedding energy in high-frequency subbands — precisely the components discarded most aggressively by JPEG and Gaussian blurring. Their *Low-frequency Image Deep Steganography* (LIDS) system extracts a secret-image feature map, then constrains its frequency distribution to low-frequency components through a dedicated frequency loss. Crucially, the container image is formed by adding this constrained feature map directly to the cover, rather than passing it through the CNN output, so the final image carries no high-frequency CNN artifacts. An auxiliary attack layer in training further hardens the embedding. LIDS achieves state-of-the-art robustness against compression and blur attacks while maintaining competitive PSNR. Its limitation is that low-frequency anchoring still yields a globally distributed embedding, providing no facility for localised damage and partial recovery.

Invertible Networks: HiNet and PRIS

Invertible neural networks (INNs) offer an alternative steganographic architecture in which the encoder is a bijective function whose exact mathematical inverse is the decoder. Jing et al. [35] introduced HiNet, the first INN-based image hiding system, achieving PSNR 37.8 dB on the standard 256×256 image-hiding task. The exact invertibility that makes INNs architecturally elegant becomes a liability under real-world channel conditions: because the decoder must precisely invert the encoder's transformation, even moderate distortion of the stego-image propagates through the inverse mapping and is magnified into large errors in the recovered representation.

Yang et al. [36] addressed this in the *Practical Robust Invertible Steganography* (PRIS) system. PRIS inserts two enhancement modules — one before and one after the extraction stage — and introduces a gradient approximation function to handle the non-differentiable rounding error that arises when 32-bit floating-point container images are saved as 8-bit integers. This rounding error, ignored by all prior INN-based systems, was shown to degrade BER significantly in practical deployment. PRIS achieves improved robustness over HiNet under Gaussian noise and JPEG compression. Nonetheless, the system retains the binary succeed-or-fail recovery model: because the entire payload is encoded holistically through the invertible mapping, partial image destruction yields an entirely corrupted output.

Attention Mechanisms and Transformer Architectures

Convolutional architectures process feature maps with local receptive fields, which may miss long-range spatial correlations useful for distributing payload energy over stable image regions. Several 2022–2024 works have introduced transformer-based watermarking and steganography encoders. Mareen et al. [37] extended HiDDeN with differentiable geometric noise layers — rotation, rescaling, translation, shearing, and mirroring — to produce the first system explicitly robust to geometric transformations within the HiDDeN family, achieving BER below 5% after combined JPEG and rotation attacks. Zhou et al. [38] demonstrated that adaptive multi-encoder–decoder pairs with a stego-aware routing mechanism achieves SSIM of 0.99 with secret-recovery accuracy exceeding 98% across a broad corruption curriculum spanning noise, compression, geometric warps, and occlusions. The routing strategy resembles a mixture-of-experts design in which different encoder–decoder specialists handle different secret-data statistical profiles.

TrustMark [39], introduced at NeurIPS, employs a GAN-based architecture with a spatio-spectral loss that jointly penalises perceptual distortion in both the pixel domain and the frequency domain. TrustMark was designed to handle arbitrary-resolution inputs and includes a companion removal network for re-watermarking scenarios. It establishes a new benchmark for robustness against standard perturbations (JPEG, Gaussian noise, brightness changes), though subsequent work has shown that all pixel-level invisible watermarks including TrustMark are highly vulnerable to diffusion model-based regeneration attacks [40].

Generative and Diffusion-Based Steganography

The most structurally significant paradigm shift of the 2021–2025 period is the move from *cover-image editing* (embed a payload into an existing photograph) to *cover-image synthesis* (generate a new image whose generative noise or latent code encodes the payload). In generative steganography the stego-image is produced by a generative model, so classical steganalysis detectors trained on edited natural photographs cannot be directly applied.

Kim et al. [41] proposed Diffusion-Stego, a training-free method that hides messages in the noise space of a diffusion model via message projection, achieving high anti-detection ability without retraining the underlying model. LDStega [42] extended this concept to Latent Diffusion Models (LDMs), encoding the secret through an encoding strategy applied during the LDM reverse (denoising) process, coupling latent-space generation with data hiding and enabling text-conditioned, controllable stego-image synthesis. RoSteALS [43] proposed latent-space steganography using frozen autoencoder representations, inheriting the generative model's imperceptibility properties without requiring encoder–decoder retraining.

These diffusion-based approaches achieve the highest anti-detection rates reported to date. However, they introduce a fundamental practical constraint: the capacity and extractability of the hidden payload depend on the architecture and sampling schedule of the underlying diffusion model. More critically for the present work, they provide *no robustness to post-generation image manipulation*: a single aggressive JPEG compression or crop of the generated stego-image destroys the implicit noise-space encoding entirely, because the decoder must invert the complete diffusion trajectory. Partial-recovery of a truncated diffusion trajectory is not currently possible.

Synthesis: Persistent Absence of Partial Recovery

Table 1.5 extends the comparison of Section 1.8.6 with representative 2021–2025 methods. The critical observation is that no system across either period provides a partial-recovery mechanism. As embedding architectures have grown more sophisticated — from convolutional encoders to invertible networks, from single-curriculum training to mixed JPEG simulation, from pixel-space editing to latent diffusion synthesis — the fundamental architectural constraint identified in Section 1.8.7 has remained unchanged: all deep decoders require the complete encoded signal and produce a binary succeed-or-fail recovery outcome. The absence of partial recovery is not an oversight but a consequence of end-to-end holistic encoding, in which the entire payload is distributed across the full image via a learned global mapping with no structural mechanism for isolating surviving fragments.

Table 1.5: Recent deep learning steganographic and watermarking methods (2021–2025). Partial Rec.: partial payload recovery from a damaged stego-image; Generative: cover is synthesised rather than edited.

Method	**Year**	**PSNR (dB)**	**BER (%)**	**Partial Rec.**
MBRS [1]	2021	36.1	0.09	No
HiNet [35]	2021	47.8	<1	No
LIDS [34]	2023	46.5	1.8	No
TrustMark [39]	2023	42.4	2-5	No
Mareen et al. [37]	2024	34.2	<5	No
PRIS [36]	2024	36.9	1.4	No
Diffusion-Stego [41]	2023	—	2.5	No
LDStega [42]	2024	—	2.5	No
Zhou et al. [38]	2025	48.2	1.7	No
Proposed (DASRS)	2025	—	—	**Yes**

1.8.6 Summary and Comparative Analysis

Table 1.6: Comparative summary of deep learning-based steganographic methods (2018–2024) alongside selected classical baselines. Performance figures are indicative for 30-bit payloads under JPEG QF = 70. BER: Bit Error Rate; PSNR: Peak Signal-to-Noise Ratio.

Method	**Year**	**PSNR (dB)**	**BER (%)**	**Partial Rec.**
DCT mid-freq. (classical)	—	≈38	≈15	No
LSB spatial (classical)	—	≈51	≈50	No
HiDDeN [25]	2018	36.1	—	No
SteganoGAN [29]	2019	30.5	40+ (JPEG)	No
ReDMark [30]	2021	36.2	0.12	No
MBRS [1]	2021	38.1	0.9	No
UDH [44]	2020	39.8	2.2	No
Proposed framework	2025	—	—	**Yes**

The critical observation is consistent with Table 1.1: to the best of our knowledge, existing approaches rarely address explicit *partial recovery* of the hidden payload when large image portions are destroyed. All deep decoders operate in a binary succeed-or-fail mode — a fundamental architectural limitation of the encoder-decoder paradigm, not a tunable parameter.

1.8.7 The Thesis Adopts a Classical Framework

Despite deep learning representing the current state of the art in raw imperceptibility and trained-attack robustness, this thesis adopts a classical, model-driven framework for four reasons directly related to the research objectives of Chapter 1.

Partial recovery as a first-class design objective. The primary objective is to recover *usable information* from a partially destroyed stego-image. No existing deep learning framework provides this; the encoder-decoder paradigm produces a holistic encoding of the entire payload, and the decoder requires the full encoded signal. Classical fragmentation and erasure-coding strategies are directly applicable in a way not yet achievable with deep architectures.

Robustness to unseen compound attacks. Deep systems are robust only to attacks represented in the training noise layer. A classical system whose robustness derives from structural payload properties (redundancy, distributed embedding, erasure coding) is robust by construction to any attack within its erasure budget, regardless of whether that attack was anticipated during design.

Interpretability and analytical guarantees. A classical framework admits formal analysis: given a specified attack model, one can compute the probability that a given fragment survives. Deep learning systems provide no such guarantees; their robustness is empirical and may degrade unpredictably.

Modularity. Individual components (ECC scheme, fragmentation strategy, embedding domain, embedding strength) can be adjusted independently. In a deep system, these choices are entangled within network weights and cannot be modified without retraining.

A third paradigm — hybrid steganography — has emerged recently and is reviewed in Section 1.9. The thesis positions DASRS as a classical framework, but the comparison with hybrid methods in Chapter 4 demonstrates that structural payload design achieves partial-recovery guarantees that hybrid architectures currently do not offer.

In summary, this thesis positions itself within classical robust steganography while drawing structural inspiration from deep learning — in particular, the idea that robustness is best achieved through a system-level design philosophy. 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.

1.9 Hybrid Steganography: Bridging Classical and Deep Learning

The preceding sections have surveyed classical steganographic methods — spatial and transform domain techniques with hand-engineered embedding rules — and deep learning approaches that replace these rules with end-to-end trainable encoder–decoder networks. A third paradigm has emerged in recent years that deliberately combines elements from both worlds: *hybrid steganography*. These systems retain the interpretability, analytical tractability, and low computational cost of classical frameworks while leveraging deep learning for adaptive region selection, content-aware embedding

strength adjustment, or post-processing refinement. This section reviews the principal hybrid architectures published between 2021 and 2025, with particular attention to their robustness properties and their relationship to the payload-survivability problem that motivates this thesis.

1.9.1 The Hybrid Design Philosophy

Hybrid steganography is motivated by a straightforward observation: neither pure classical nor pure deep learning methods fully resolve the steganographic trilemma (Section **??**). Classical techniques offer modularity and interpretability but lack adaptability to image content; deep learning methods achieve content adaptability but sacrifice interpretability, require large training datasets, and suffer from distribution-shift vulnerability (Section **??**). Hybrid approaches attempt to capture the best of both paradigms by assigning complementary roles to classical and learned components.

The typical hybrid pipeline follows one of three architectural patterns:

1. **CNN-guided classical embedding:** A lightweight convolutional network analyses the cover image and produces an embedding map or region mask; a classical transform-domain algorithm (DCT, DWT, DFT) then embeds data according to this guidance.

2. **Classical-domain deep refinement:** Data is embedded using a classical method; a generative network (GAN or autoencoder) then post-processes the stego-image to reduce statistical artifacts and improve imperceptibility.

3. **Multi-stage fusion:** Classical transforms provide frequency-domain representations that serve as input features to a deep network, which learns to embed in the transformed coefficients rather than in raw pixels.

All three patterns share a common structural advantage: the deep learning component handles *where* and *how much* to embed, while the classical component determines *how* the embedding is physically realised. This separation of concerns preserves the robustness guarantees of the transform domain — resistance to JPEG quantisation, spatial localisation, compatibility with compression standards — while allowing the system to adapt to local image statistics.

1.9.2 CNN-Guided DCT Embedding

A representative hybrid architecture in this category was proposed by Ahmad et al. (2024), who introduced *CNN-DCT Steganography* for secure cloud storage applications [45]. The system operates in two stages. First, a deep CNN analyses the cover image and extracts high-level feature maps that identify texture-rich, high-entropy regions suitable for embedding. Second, the selected regions are transformed via block DCT and secret data is embedded into mid-frequency coefficients using a quantisation-based rule. The CNN does not perform the embedding itself; it merely provides a *spatial attention map* that constrains the classical DCT embedding to blocks where coefficient modifications are least perceptible.

Experimental evaluation on 500 high-resolution images reported a PSNR of 42.5 dB, SSIM of 0.98, and BER of 0.028 (2.8%) under normal conditions [45]. The method demonstrated robustness against common image transformations, though explicit attack-by-attack BER figures were not published. The computational cost is dominated by the CNN forward pass (2.3 s per image on standard hardware), making it feasible for cloud-based deployment but an order of magnitude slower than purely classical methods.

A related approach by Al-Rawashdeh et al. (2025) integrates edge detection with CNN-based adaptive embedding [46]. Edge pixels are first identified using Canny or Sobel operators; a CNN then refines the embedding strength within edge regions based on local texture complexity. The method achieves PSNR up to 39.85 dB (Sobel-guided) and 33.08 dB (Canny-guided), with SSIM exceeding 0.995. Capacity reaches 8 bpp, significantly higher than transform-only methods, but robustness

evaluation is limited to JPEG compression, Gaussian noise, and salt-and-pepper noise. Steganalysis resistance is reported via Xu-Net and Ye-Net detection accuracies of 0.45–0.67, indicating moderate resistance.

These CNN-guided approaches share a limitation relevant to this thesis: the CNN provides adaptive *spatial* selection but does not alter the *structural* properties of the payload. The message remains a single, undivided bit stream with no fragmentation, parity redundancy, or partial-recovery mechanism. Consequently, if the DCT embedding in a guided region is destroyed by attack, the entire message is lost — the same binary succeed-or-fail behaviour observed in pure classical and pure deep learning systems.

1.9.3 DCT–GAN Hybrid Frameworks

The second architectural pattern — classical embedding followed by deep refinement — is exemplified by the DCT–GAN framework of Malik et al. (2025) [47]. In this system, secret data is first embedded into the LSBs of DCT coefficients using a conventional frequency-domain technique. The resulting stego-image is then passed through a GAN comprising an encoder (nine residual blocks), a decoder, and a discriminator. The generator learns to refine the stego-image so that its statistical distribution matches that of natural images, thereby reducing detectability. The encoder–decoder module additionally provides a denoising function that compensates for mild distortions.

The DCT–GAN hybrid achieves impressive imperceptibility metrics: PSNR of 58.27 dB, SSIM of 0.982, and MSE of 93.30% on the ImageNet dataset [47]. Reconstruction accuracy against steganalysis networks reaches 96.2% (Xu-Net) and 95.7% (SR-Net). However, robustness against explicit signal-processing attacks is not quantified in terms of BER per attack type; the reported BER of 0.15 (15%) appears to be an average over unstated conditions. The embedding capacity is low at 0.039–0.079 bpp, reflecting the conservative embedding necessitated by the GAN refinement stage.

A critical observation is that the GAN refinement in this architecture operates *after* embedding. It can mask visual artifacts but cannot recover data that has been physically destroyed by compression, cropping, or noise. The robustness bottleneck remains the classical DCT embedding stage; the GAN improves imperceptibility and steganalysis resistance but does not enhance payload survivability under attack.

1.9.4 Wavelet–CNN Fusion Methods

The third pattern — deep networks operating on classical transform coefficients — is represented by DeepWaveletFusion and its successor DeepWaveletFusionToo (2025) [48]. These methods train convolutional networks on DWT-transformed images rather than on raw pixels. The wavelet decomposition provides a multi-resolution representation that separates approximation and detail sub-bands; the CNN learns to embed secret data into these sub-band coefficients while preserving both stego-image quality and secret recoverability.

DeepWaveletFusionToo achieves a PSNR of approximately 40 dB and a secret-recovery similarity of 93% on custom datasets [48]. The method outperforms earlier deep steganography systems such as DeepStego (PSNR 28.69 dB, similarity 73%) and is competitive with SteGuz (PSNR 44.33 dB, similarity 93%) while offering greater architectural flexibility. However, robustness against common image processing attacks — JPEG compression, resizing, filtering — is not evaluated in the published work, and the method inherits the binary succeed-or-fail recovery model of its deep learning substrate.

A reversible hybrid framework combining the Radon Transform (RT) with the Integer Lifting Wavelet Transform (ILWT) was proposed in 2025 [49]. This system embeds data into the middle bit planes of high-frequency ILWT sub-bands (LH, HL, HH) using arithmetic coding. The RT provides geometric robustness against rotation, scaling, and translation without requiring inverse transformation, while ILWT guarantees exact reversibility through integer coefficients. The method achieves PSNR of 56.22 dB, SSIM of 0.999, and BER of 0.156 under mild conditions. Notably, it demonstrates strong robustness against cropping: 100% data recovery after 10% cropping and 98.5%

after 15% cropping — a partial-recovery behaviour that approaches the PRR metric introduced in this thesis, though without explicit fragment-level granularity.

1.9.5 Lightweight Hybrid Models: StegTransX

Among the most practically oriented hybrid systems is StegTransX (2025), a lightweight encoder–decoder network that explicitly targets JPEG compression resistance [50]. StegTransX combines TransX blocks (convolution + attention) with Channel–Spatial Collaborative Fusion (CSCF) for feature mixing, and introduces a dedicated JPEG compression simulation module during training. The architecture is designed to be parameter-efficient while maintaining high capacity and imperceptibility.

Experimental results on DIV2K, COCO, and ImageNet demonstrate PSNR improvements of more than 4 dB over prior state-of-the-art methods for single-image hiding, with competitive multi-image hiding performance [50]. The JPEG attack module enables reliable secret extraction after compression, though explicit BER figures per quality factor are not reported. StegTransX represents a turning point in practical deep steganography by balancing capacity, imperceptibility, and compression robustness within a lightweight design. However, like all encoder–decoder systems, it provides no partial-recovery mechanism: if the stego-image is partially destroyed, the decoder fails holistically.

1.9.6 Summary and Positioning Relative to DASRS

Table 1.7 summarises the key hybrid methods reviewed in this section alongside the classical and deep learning baselines previously discussed.

Table 1.7: Comparison of hybrid steganography methods (2021–2025) with classical and deep learning paradigms. Key: PSNR in dB; BER in %; Capacity in bpp; Partial Rec. = partial recovery capability.

Method	Year	Type	PSNR (dB)	BER (%)	Capacity (bpp)	Partial Rec.
DCT baseline (this thesis)	—	Classical	37.68	≈15 (JPEG QF 70)	Low	No
DWT baseline (this thesis)	—	Classical	43.27	≈15 (JPEG QF 70)	Low	No
HiDDeN [25]	2018	Deep learning	36.1	≈4	Medium	No
MBRS [1]	2021	Deep learning	38.1	0.09	Medium	No
CNN-DCT [45]	2024	Hybrid (CNN + DCT)	42.5	2.8	Medium	No
Edge+CNN [46]	2025	Hybrid (Edge + CNN)	39.85	0.32–0.47	8.0	No
DCT–GAN [47]	2025	Hybrid (DCT + GAN)	58.27	15 (avg.)	0.039–0.079	No
RT–ILWT [49]	2025	Hybrid (RT + ILWT)	56.22	15.6	0.48	Limited
DeepWaveletFusionToo [48]	2025	Hybrid (DWT + CNN)	≈40	N/R	Medium	No
StegTransX [50]	2025	Hybrid (CNN + JPEG sim.)	>40	N/R	High	No
DASRS (proposed)	**2025**	**Classical (multi-domain)**	**33.05**	**0–4.85**	**Low**	**Yes**

Several patterns emerge from this comparison. First, hybrid methods consistently achieve higher PSNR than pure classical methods, often rivalling or exceeding deep learning systems, because the deep component optimises embedding locations to minimise perceptual distortion. Second, BER figures for hybrid systems are generally lower than classical baselines but comparable to deep learning methods; however, most hybrid papers do not report per-attack BER breakdowns, making rigorous robustness comparison difficult. Third, and most importantly for this thesis, *no hybrid method provides a structured partial-recovery mechanism*. Whether the architecture is CNN-guided DCT, DCT–GAN refinement, or wavelet–CNN fusion, the payload remains a single, monolithic bit stream. Recovery is binary: either the complete message is extracted, or nothing meaningful is recovered.

The DASRS framework proposed in this thesis differs fundamentally from the hybrid paradigm. Rather than adding a deep learning component to a classical embedding pipeline, DASRS addresses robustness through *structural payload design*: fragmentation, cross-fragment parity, and multi-domain redundant embedding. This is a classical-system approach that achieves partial recovery without neural networks, training data, or GPU acceleration. The hybrid methods reviewed here

demonstrate that combining classical and deep learning elements can improve imperceptibility and steganalysis resistance; however, they do not solve the payload-survivability problem that motivates this work. Chapter 4 includes a direct performance comparison between DASRS and representative hybrid techniques.

Chapter 2

Proposed Framework and Implementation

The framework proposed in this work, designated DASRS (Damage Aware Self-Recovering Steganography), emerged from a straightforward but consequential observation: the steganographic techniques reviewed in Chapter 1 are designed to make data invisible, not to make it survive. When a stego-image is altered by compression, filtering, or geometric transformation, the embedded payload is damaged or destroyed, and no mechanism exists that would allow even a partial recovery of the original message. DASRS addresses this gap by treating the embedded payload as a resilient data structure rather than a raw bit sequence, and by distributing that structure across multiple independent embedding domains. This chapter describes both the design rationale behind this approach and its complete implementation, covering the payload architecture, the multi-domain embedding strategy, the extraction cascade, and the self-recovery mechanism.

2.1 Design Motivation

Every design decision in DASRS is a direct response to a specific, identifiable failure mode in existing steganographic methods. The failures described in this section were observed in the baseline experiments conducted as part of this study and are consistent with the theoretical limitations documented in the literature surveyed in Chapter 1.

2.1.1 Limitations of Existing Steganographic Approaches

The methods surveyed in Chapter 1 span a wide technical range, from direct pixel manipulation in the spatial domain to coefficient modification in the DCT, DWT, and DFT frequency representations of the image. Despite this diversity, they share a structural characteristic that places a hard ceiling on their robustness: the secret message is treated as a single, undivided data stream. Recovery is consequently binary — either the complete message is extracted correctly, or nothing meaningful is recovered at all. There is no provision for partial retrieval, and no accommodation for the scenario in which only a fraction of the embedded data survives a given image transformation.

Three distinct deficiencies follow from this structural property. The first is

Deficiency 1 — Domain-specific fragility . Each embedding domain provides protection against a particular class of distortions but is itself vulnerable to others. DCT mid-frequency embedding tolerates JPEG compression but degrades under additive noise, where small random perturbations repeatedly shift coefficient values across the decision boundaries that carry the embedded bits. DWT embedding in the approximation subbands resists moderate noise but is sensitive to low-pass filtering and aggressive compression. Spatial-domain QIM tolerates isolated bit errors through majority voting but collapses under any lossy compression that modifies pixel values beyond the

quantisation step. No single domain offers broad-spectrum robustness [10], and a method confined to one domain inherits that domain's vulnerability in full.

Deficiency 2 — Absence of payload structure . When the message is embedded as a raw bit sequence with no internal organisation, any corruption in a critical region of the stego-image can render the entire message unrecoverable. The damage need not be extensive: a 5 % border crop — an operation applied automatically by a significant fraction of image-sharing platforms — removes all embedding capacity from the cropped region. If that region happens to carry the opening bits of the message, which in sequential embedding schemes it typically does, the remaining content is rendered meaningless regardless of how well it survived. The payload has no capacity to redistribute meaning around a gap [8].

Deficiency 3 — Extraction without damage awareness . Standard extraction procedures assume, either explicitly or implicitly, that the stego-image is a recognisable descendant of the original. The extraction algorithm is applied once, uniformly, across the full image, and the result is accepted or discarded as a whole. There is no step that detects whether a particular region has been corrupted, no fallback that attempts an alternative if the primary extraction fails, and no reconstruction step that could recover meaning from the fragments that did survive. When the image is clean this simplicity is an asset; when the image has been subjected to compound or severe transformations it becomes the direct cause of total failure.

2.1.2 The Need for a Payload-Survivability-Oriented Framework

The three deficiencies above share a common root: robustness is treated as a property of where data is hidden rather than of how the data itself is organised. The design assumption is that a carefully chosen embedding region, sufficiently stable under the expected attacks, will passively preserve the hidden information. The payload is inert — it contributes nothing to its own survival.

A different model is possible. If the message is transformed into a structured, redundant object before embedding — divided into labelled fragments, protected by error-correcting codes, bound by a parity relationship that permits reconstruction from incomplete recovery, and distributed across embedding locations that fail under different conditions — then the survival of any subset of fragments is sufficient to yield a partial recovery, and the survival of enough fragments makes a complete recovery possible even after attacks that destroy substantial portions of the image. Robustness becomes a property of the payload structure, not just of the embedding domain.

This reorientation from hiding-centred to survivability-centred design is the founding motivation for DASRS. The system makes no claim to hide data more invisibly than existing methods, and it accepts a modest reduction in imperceptibility as the direct cost of structural resilience. What it offers in exchange is measurable: given a hostile channel that applies arbitrary distortions to the stego-image, a larger fraction of the embedded message will survive than would survive with any single-domain method operating under identical conditions.

2.1.3 Core Design Principles of DASRS

Three design principles govern the architecture of DASRS, each mapping directly onto one of the deficiencies identified in Section 2.1.1.

1: multi-domain distributed redundancy . Every fragment of the secret message is embedded simultaneously into four independent domains, each operating on a distinct channel of the YCbCr colour space and employing a different transform. The independence of the colour channels guarantees that an attack targeting one domain does not, in general, affect the others. An operation that equalises the luma histogram, for instance, modifies the Y channel exclusively and leaves Cr and Cb

unchanged. The robustness contribution of each domain is therefore additive: at least one domain is likely to survive any attack short of complete image destruction.

2: structured fragmentation with inter-fragment parity . The message is divided into six equal data fragments, and a seventh fragment is constructed as their byte-wise XOR. This arrangement, analogous in principle to the parity scheme used in RAID-5 distributed storage, allows any single lost data fragment to be reconstructed exactly from the surviving five and the parity fragment, without approximation or probabilistic estimation [51]. Within each fragment, a Reed-Solomon encoder provides a second, finer-grained layer of protection against byte-level errors, independently of the cross-fragment parity mechanism [52].

3: damage-aware cascade extraction . The extraction stage does not assume that the stego-image is intact. Each fragment is sought across all four embedding domains in a fixed order of decreasing resilience, and is accepted only after passing a three-condition validation test. The system records which fragments were recovered and which were lost, then invokes the XOR parity reconstruction when the conditions for exact recovery are satisfied. The final output is the recovered message together with a numerical score that quantifies how much of the original message was successfully retrieved.

2.2 DASRS Framework Overview

With the design principles established, this section presents the DASRS system at the architectural level before the individual components are analysed in detail in the sections that follow.

2.2.1 The Three Pillars of the Framework

The DASRS architecture is organised into three layers that correspond one-to-one with the three design principles and that operate without direct coupling between them.

The **structured payload layer** transforms the raw message into an error-protected, parity-equipped set of binary fragments before any embedding takes place. It handles fragmentation, CRC-16 integrity checksums per fragment, Reed-Solomon encoding, and the construction of the XOR parity fragment. This layer has no knowledge of the domains into which the fragments will be embedded; it produces a collection of bit sequences and passes them to the layer above.

The **multi-domain embedding layer** receives these bit sequences and distributes them across the cover image. It is responsible for colour space conversion, zone partitioning to prevent any two fragments from sharing embedding blocks, and the execution of four independent embedding procedures — one per domain — each operating on a different channel and at a different frequency scale. This layer treats the bit sequences as opaque binary data; it has no knowledge of what they represent.

The **damage-aware extraction layer** operates at the receiver on what may be a degraded copy of the stego-image. It performs cascade domain testing, validates each candidate fragment through a triple check combining Reed-Solomon decoding, CRC verification, and identity confirmation, and invokes the parity reconstruction mechanism when conditions are met. This layer interacts with the payload layer only through the binary content of the recovered fragments; it does not know or need to know how those fragments were embedded.

The separation between these layers is intentional. The specific domain configuration used in this implementation — four channel-transform pairs chosen for their complementary robustness profiles — is one concrete realisation of the framework. A different domain set could be substituted without altering the payload or extraction logic, provided the chosen domains remain statistically independent under the attacks of interest. The generality of the framework is therefore broader than the specific implementation evaluated here.

2.2.2 Sender-Side Pipeline

The sender-side pipeline transforms a plaintext message into a stego-image in six sequential stages, illustrated in Figure 2.1.

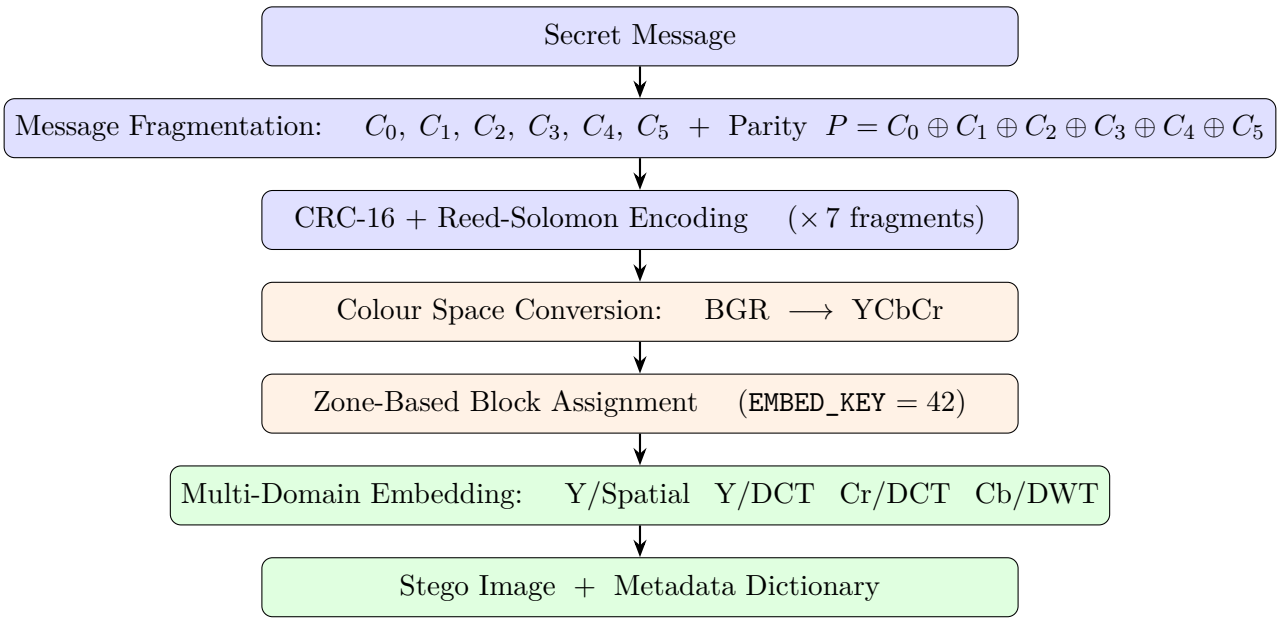


Figure 2.1: DASRS sender-side processing pipeline. Blue nodes represent payload preparation stages; orange nodes represent image-level preparation; green nodes represent the embedding and output stages.

In the first stage, the message is divided into six equal data chunks $C_0, C_1, C_2, C_3, C_4, C_5$, and a seventh parity chunk P is computed as their byte-wise XOR. Each chunk is then assembled into a structured fragment containing the chunk data and a CRC-16 checksum of that chunk. This checksum allows the extraction stage to verify the integrity of the decoded content independently in each domain. A Reed-Solomon encoder then appends sixteen parity bytes to each fragment, producing a binary sequence that can correct up to eight byte-level errors per fragment before the cross-fragment parity mechanism is even considered. The seven encoded fragments are the atomic units of the DASRS payload from this point forward.

In the second stage, the cover image is converted from BGR to the YCbCr colour space. This separates the image into a luma channel Y and two chrominance channels Cr and Cb. The three channels are processed independently from this point onward; no operation on one channel modifies the values of another. This independence property is central to the multi-domain redundancy argument and is discussed in detail in Section 2.4.

In the third stage, the image is partitioned into non-overlapping zones. Each of the seven fragments is assigned an exclusive zone within each of the four embedding domains, for a total of twenty-eight zone assignments with no shared blocks between any two fragments in any domain. A pseudo-random number generator seeded with the fixed key `EMBED_KEY = 42` shuffles the order of blocks within each zone, introducing controlled unpredictability without violating the exclusivity constraint that prevents fragments from overwriting one another — a failure mode that affected earlier versions of the system and is documented in Section 2.4.3.

In the fourth and final stage, each fragment is embedded into its assigned zone in all four domains simultaneously. The embedding procedures and their mathematical formulations are described in detail in Section 2.4. The output is the stego-image together with a metadata dictionary recording the structural parameters required for extraction: fragment count, message byte length, chunk size, bit counts per fragment, and zone boundary information. This metadata must accompany the stego-image to the receiver; it carries no information about the message content itself.

2.2.3 Receiver-Side Pipeline

The receiver-side pipeline operates on a potentially degraded copy of the stego-image. Its design premise is that some fragments will have survived intact, some may have been partially corrupted, and some may be entirely unrecoverable — and that a useful output can still be produced from whatever subset of the payload remains. The pipeline is illustrated in Figure 2.2.

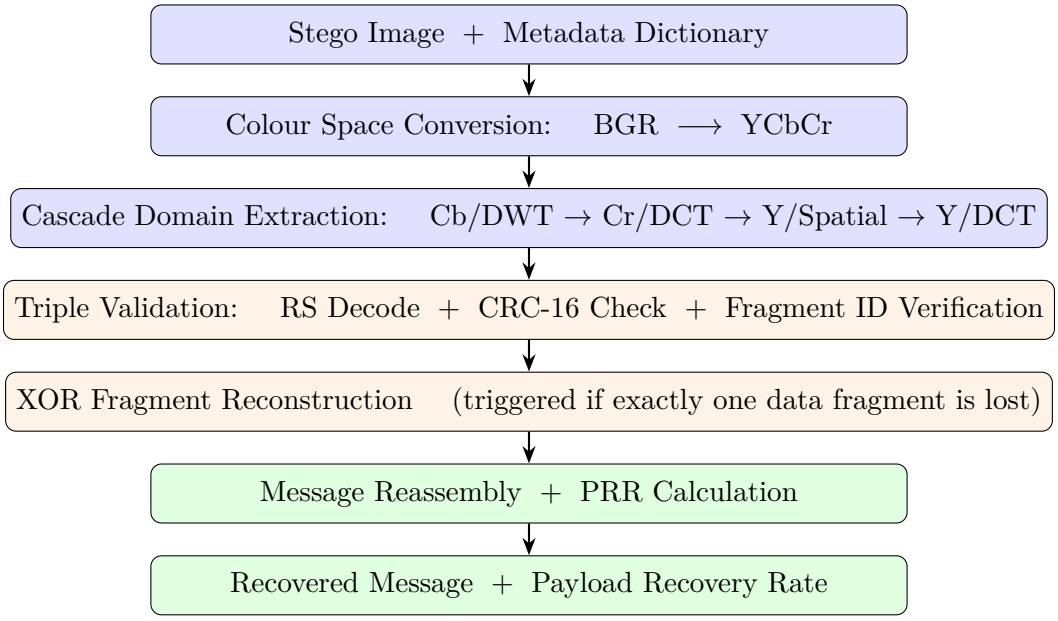


Figure 2.2: DASRS receiver-side processing pipeline. Blue nodes represent extraction stages; orange nodes represent validation and reconstruction; green nodes represent reassembly and output.

The first two stages mirror the sender: the stego-image is converted to YCbCr, yielding the same three-channel representation from which the fragments were originally embedded. If the image has been resized, the channel arrays will be of different dimensions than at embedding time; the extraction procedures accommodate this within the limits described in Section 2.4.6.

In the third stage, each of the seven fragments is sought independently across all four embedding domains in the cascade order shown in Figure 2.2. The Cb/DWT domain is tested first because it is the only domain that provides reliable extraction after resize operations: the level-2 wavelet decomposition encodes each bit at a coarser spatial scale that bilinear interpolation leaves largely intact. The Cr/DCT domain is tested second because it resides on a colour channel unaffected by the majority of luma-modifying operations. The Y/Spatial domain is tested third because its majority-vote extraction aggregates the signal from sixty-four pixels per bit, attenuating the effect of isolated pixel errors. The Y/DCT domain — the most information-dense but also the most sensitive to distortions that affect coefficient magnitudes — is reserved as a last resort.

For each domain, the extraction reads the bits from the fragment's assigned zone, attempts Reed-Solomon decoding, and if decoding succeeds, verifies the CRC-16 checksum and the fragment identity marker. All three conditions must hold for the fragment to be accepted. A failure at any point causes the system to advance to the next domain in the cascade without recording a partial result; fragments are either fully accepted or not accepted at all. A fragment that fails in all four domains is recorded as lost.

After the cascade has been applied to all seven fragments, the reassembly stage examines the recovery state. If all six data fragments were validated, the message is assembled directly from their data chunks in order. If exactly one data fragment is missing and the parity fragment was recovered, the reconstruction mechanism is triggered: the missing chunk is computed exactly as the byte-wise XOR of the parity chunk and the five surviving data chunks. This operation introduces no approximation — it is an algebraic identity, not an estimation. If two or more data fragments are missing, reconstruction is not possible, and the message is returned in whatever partial form

the recovered fragments allow. In all cases, the Payload Recovery Rate is computed and returned alongside the recovered message. The formal definition of this metric is given in Section 2.5.5.

2.3 Structured Payload Design

The most consequential design choice in DASRS is not the selection of embedding domains — it is the transformation applied to the message before any embedding takes place. Classical steganographic methods pass the message bits, possibly after simple binary serialisation, directly into the embedding function. The payload is therefore a featureless sequence: any damage to the bit stream at extraction yields an output that is either wrong in unpredictable ways or undecodable entirely. DASRS takes a different approach. Before the first pixel or coefficient is modified, the message is converted into a self-describing, error-protected, parity-equipped collection of independent fragments. Each fragment carries enough structural information to be validated on its own, and the collection as a whole is organised so that any single lost member can be reconstructed from the remaining five. This section describes that transformation in detail.

2.3.1 Message Fragmentation Strategy

The starting point is the observation that a message embedded as a single continuous block places all of its information at risk whenever any part of the image that holds it is damaged. The natural counter-measure is division: if the message is split into several independent pieces and those pieces are embedded in regions that are likely to fail under different attack conditions, then a distortion severe enough to destroy one piece will in many cases leave the others unaffected. This is not a new idea in the broader field of data communications — systematic distributed storage and erasure-coded transmission have exploited the same principle for decades [51] — but its application to image steganography as a first-class design element, rather than an incidental detail, distinguishes DASRS from the methods discussed in Chapter 1.

Fragment Count and Chunk Size Calculation

The message is divided into $N_{\text{data}} = 6$ data fragments of equal size. The choice of six reflects a deliberate balance between recovery granularity and embedding overhead. Fewer fragments reduce the overhead of the parity scheme and increase the capacity per fragment but coarsen the partial-recovery resolution: with two fragments a failure is either total or half the message survives, whereas with six fragments the PRR grades are $\frac{1}{6} \approx 17\,\%$, $\frac{2}{6} \approx 33\,\%$, $\frac{3}{6} = 50\,\%$, $\frac{4}{6} \approx 67\,\%$, $\frac{5}{6} \approx 83\,\%$, and $100\,\%$, allowing fine-grained characterisation of damage extent. More fragments would further increase this granularity but raise the total bit count to be embedded, reducing the range of cover images that satisfy the minimum capacity requirement and increasing the perceptual distortion on the cover image.

The message is first encoded as a UTF-8 byte string of length ℓ bytes. If ℓ is not exactly divisible by N_{data}, zero bytes are appended until the total length is a multiple of six. The chunk size is then:

$$s = \left\lceil \frac{\ell}{N_{\text{data}}} \right\rceil \tag{2.1}$$

and the six data chunks $C_0, C_1, C_2, C_3, C_4, C_5$ each contain exactly s bytes, with the padding zeros occupying the tail of C_5 when necessary. As a concrete example, the test message used in this study, *"Hello, this is a secret message hidden with DASRS!"*, is 49 bytes long. Padding raises this to 54 bytes ($\lceil 49/6 \rceil = 9$, $6 \times 9 = 54$), yielding six chunks of 9 bytes each.

A seventh fragment P, the XOR parity fragment, is constructed from these six chunks as described in Section 2.3.2, bringing the total number of embedded fragments to $N_{\text{fragments}} = 7$.

2.3.2 Inter-Fragment Redundancy via XOR Parity

Dividing the message into independent fragments solves the problem of localised damage but introduces a new vulnerability: if one fragment is lost, the corresponding portion of the message is lost with it. To address this, a seventh fragment is constructed whose sole purpose is to provide enough information to reconstruct any single missing data fragment. This arrangement is structurally equivalent to the parity mechanism used in RAID-5 distributed disk storage [51], adapted here to operate at the byte level on steganographic payload chunks.

Parity Fragment Construction

Once the six data chunks have been assembled, the parity chunk P is computed by initialising a zero-filled buffer of length s bytes and folding each data chunk into it through byte-wise exclusive OR. The parity fragment is then serialised and encoded identically to the data fragments — it carries the same wire format, the same CRC-16 checksum, and the same Reed-Solomon protection. At the embedding stage, the parity fragment is treated as a seventh payload element with no special status; the embedding pipeline does not distinguish between data fragments and the parity fragment in any way. This simplifies the implementation and guarantees that the parity fragment benefits from the same four-domain redundancy as every other fragment.

XOR Parity Computation

For a chunk size of s bytes, the parity chunk is defined byte-position by byte-position as:

$$P[j] = C_0[j] \oplus C_1[j] \oplus C_2[j] \oplus C_3[j] \oplus C_4[j] \oplus C_5[j], \quad j = 0, 1, \ldots, s-1 \tag{2.2}$$

where $\oplus$ denotes the bitwise exclusive OR operation. If during extraction exactly one data fragment C_i is found to be irrecoverable, and the parity fragment P is available, the lost chunk is recovered exactly by applying the same XOR identity to the surviving pieces:

$$C_i[j] = P[j] \oplus \bigoplus_{\substack{m=0 \\ m \neq i}}^{5} C_m[j], \quad j = 0, 1, \ldots, s-1 \tag{2.3}$$

The correctness of Equation (2.3) follows directly from the associativity and self-inverse property of XOR: since $P[j] = C_0[j] \oplus \cdots \oplus C_5[j]$, XOR-ing $P[j]$ with the five known chunks cancels all terms except $C_i[j]$, leaving the lost byte recovered exactly. There is no approximation, no probabilistic estimation, and no dependence on the content of the recovered bytes. The reconstruction is as reliable as the surviving fragments themselves.

Limitation: Single-Fragment Recovery Only

The XOR scheme described above is capable of reconstructing at most one lost data fragment. If two or more data chunks are simultaneously irrecoverable, Equation (2.3) cannot be applied because at least one unknown remains on the right-hand side. In this case, the system assembles the message from whatever data fragments were successfully recovered and leaves the missing portions as null bytes, reducing the Payload Recovery Rate proportionally.

This is a known limitation shared with RAID-5, where simultaneous failure of two disks also exceeds the parity scheme's corrective capacity. In the steganographic context, the constraint is mitigated by two design features: the four-domain embedding, which makes it unlikely that any two fragments will simultaneously fail across all four of their respective embedding domains, and the Reed-Solomon encoding within each fragment, which handles partial bit corruption before the fragment-level loss decision is even made. The practical impact of this limitation on the experimental results is analysed in Chapter 4.

2.3.3 Fragment Wire Format

Each fragment, whether data or parity, is serialised into a fixed-structure byte sequence before Reed-Solomon encoding. This wire format is identical for all seven fragments and determines the total bit count that must be accommodated within each fragment's embedding zone.

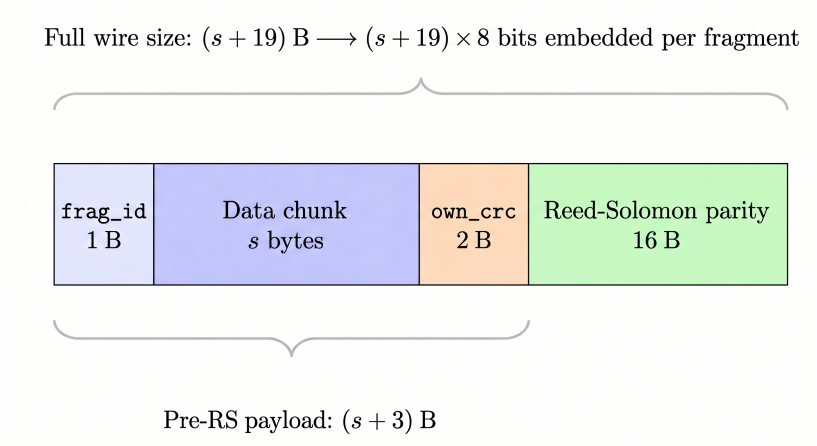


Figure 2.3: DASRS fragment wire format showing the four fields: fragment ID, data chunk, CRC-16 checksum, and Reed–Solomon parity bytes.

The wire format, shown in Figure 2.3, consists of four fields arranged in the following order. The **fragment identifier** occupies the first byte and records the index of the fragment in the range $0 \leq \texttt{frag\_id} \leq 6$, with index 6 reserved for the parity fragment. The **data chunk** follows immediately and contains the s message bytes assigned to this fragment, or the XOR parity bytes if $\texttt{frag\_id} = 6$. The **CRC field** records the CRC-16 checksum of this fragment's own data chunk (`own_crc`), allowing the extraction stage to verify the integrity of the decoded content. The final field, appended by the Reed-Solomon encoder, contains sixteen bytes of error-correction symbols.

The total wire size is therefore $(s + 19)$ bytes per fragment, or $(s + 19) \times 8$ bits. For the 9-byte chunk of the test message, each fragment occupies $(9 + 19) \times 8 = 224$ bits. It is this bit count that determines the minimum embedding capacity required per zone in each domain, and it governs the minimum image size at which the system can operate, as discussed in Section 2.4.6.

2.3.4 CRC-16 Integrity Verification

Each fragment carries a CRC-16 checksum (`own_crc`) that covers its data chunk. This checksum serves as an independent integrity guard: even when Reed-Solomon decoding succeeds, a subtle corruption that falls within the RS correction budget may produce an incorrect but syntactically valid codeword. The CRC check catches such cases by comparing the recomputed checksum of the decoded chunk against the stored `own_crc` value. A mismatch causes the candidate to be rejected immediately, and the cascade advances to the next domain.

The checksum function used throughout is CRC-16/CCITT-FALSE, defined by the generator polynomial:

$$G(x) = x^{16} + x^{12} + x^5 + 1 \tag{2.4}$$

with an initialisation value of `0xFFFF` and no final XOR or bit reversal. This variant is widely used in communications protocols for its strong error-detection properties with 16-bit overhead: it detects all single and double bit errors, all odd numbers of bit errors, all burst errors of length 16 or fewer bits, and the vast majority of longer burst errors [53]. For the fragment lengths used here — at most a few hundred bits — the probability of an undetected error given that a corruption occurred is $2^{-16} \approx 1.5 \times 10^{-5}$, which is negligible relative to the other sources of uncertainty in the system.

At extraction, a candidate fragment is accepted only when all three of the following conditions hold simultaneously:

$$\texttt{RS\_decode}(\mathbf{b}) = \text{success} \ \wedge \ \texttt{crc16}(C_i) = \texttt{own\_crc} \ \wedge \ \texttt{frag\_id} = i_{\text{expected}} \tag{2.5}$$

where $\mathbf{b}$ is the extracted bit sequence and i_{expected} is the fragment index currently being sought. Failure of any single condition causes the extraction to advance to the next domain in the cascade without recording a result. This strictness is intentional: a fragment accepted with an incorrect identity or corrupted data would introduce a silent error into the reassembled message, which is a worse outcome than marking the fragment as lost and invoking the parity reconstruction mechanism.

2.3.5 Reed-Solomon Error Correction

The CRC-16 scheme described above detects errors but cannot correct them. A fragment that fails its CRC check must be either discarded or sought in another domain. In many scenarios, however, only a few bytes within the fragment are corrupted — the kind of damage inflicted by moderate Gaussian noise or a handful of JPEG quantisation rounding events — and discarding the entire fragment on account of a small number of corrupted bytes would be unnecessarily wasteful. Reed-Solomon codes are designed precisely for this situation: they locate and correct a bounded number of corrupted symbols within a codeword, restoring the fragment to its original state before the CRC check is even applied [52].

In DASRS, each fragment is encoded using a (n, k) Reed-Solomon code over $\mathrm{GF}(2^8)$ with $t = 16$ parity symbols, where $t = n - k$. The decoder is guaranteed to correct up to $\lfloor t/2 \rfloor = 8$ byte errors per fragment, regardless of their position within the codeword. If the number of corrupted bytes exceeds eight, the decoder will either return a decoding failure — in which case the fragment is passed to the next cascade domain — or, in rare cases of very heavy corruption, produce an incorrect correction that the CRC check will subsequently reject. The eight-byte correction capacity was chosen as a practical balance: it is large enough to absorb the byte-level damage inflicted by moderate compression and noise, yet it keeps the RS parity overhead at 16 bytes, a figure that adds less than 15 % to the wire size for the chunk sizes used in this implementation.

Two-Level Protection Architecture

The combination of intra-fragment Reed-Solomon coding and inter-fragment XOR parity creates two distinct and independent layers of protection that operate at different granularities and against different failure modes. Table 2.1 summarises their respective scopes.

Table 2.1: The two-level error protection architecture of the DASRS payload. The two layers operate independently; each addresses a category of damage that the other cannot handle.

Layer	Mechanism	Scope	Corrects
Intra-fragment	Reed-Solomon ECC	Within one fragment	Up to 8 corrupted bytes per fragment
Inter-fragment	XOR parity (RAID-5 style)	Across all fragments	Any single completely lost data fragm

The intra-fragment layer acts first, at the individual fragment level. When bits are extracted from a domain and the RS decoder is invoked, any corrupted bytes that fall within the correction are silently repaired before the fragment is presented to the CRC checker. The fragment either emerges from RS decoding in a corrected state or the decoding fails outright. In the former case, the CRC check operates on data that is already restored to its original form.

The inter-fragment layer acts second, after the cascade has determined which data fragments survived intact and which did not. It does not concern itself with byte-level errors — those are handled by the RS layer — but only with the binary outcome of whether each fragment was recovered at all. If the cascade returns three validated data fragments and a validated parity fragment, the XOR reconstruction fires and restores the fourth. The two layers therefore address complementary failure modes: partial corruption within a surviving fragment, and total loss of an entire fragment.

This two-level structure means that an attack must simultaneously overcome the RS correction capacity *and* destroy more than one complete fragment in order to cause unrecoverable data loss. In practice, the attacks that destroy entire fragments — severe cropping, for example — tend to do so in a spatially localised way that affects all four domains of one fragment simultaneously, which is why cropping remains a shared failure mode. Attacks that introduce distributed byte-level corruption, such as moderate Gaussian noise, typically stay within the RS correction budget and are resolved entirely by the intra-fragment layer, leaving the inter-fragment parity layer uninvoked.

2.4 Multi-Domain Redundant Embedding

The structured payload produced in Section 2.3 is a collection of seven independent bit sequences, each protected by Reed-Solomon coding and bound to the others through a parity relationship. The question this section addresses is where and how those bit sequences are written into the cover image. The answer has two parts: the choice of domains in which to embed, and the engineering decisions that make the domains genuinely independent from one another. A domain that is merely distinct in name but correlated in its response to attacks provides no additional resilience — it is independent failure modes, not independent algorithms, that matter.

2.4.1 The Concept of Domain-Level Redundancy

A single embedding domain can be understood as a channel through which the hidden bits survive the journey from sender to receiver. Every channel has a characteristic set of attacks that it tolerates and a characteristic set that destroys it. DCT mid-frequency embedding, for instance, resists JPEG compression by design: the quantisation step at a given coefficient position is known in advance, and the embedding strength can be set above it. The same domain is sensitive to additive noise, which randomly perturbs the coefficient values that carry the comparison signal. DWT subband embedding at coarse scales resists interpolation-based resize operations but is vulnerable to the low-pass smoothing effect of Gaussian blur applied at sufficient strength. Spatial-domain quantisation index modulation accumulates votes across an entire block of pixels, making it tolerant of isolated impulse noise, yet it degrades under any operation that systematically shifts pixel intensities.

No single domain is universally robust, and combining domains that fail under different conditions directly addresses this constraint [10]. If each fragment is embedded into four domains simultaneously, and if those domains fail independently under the attacks of interest, then a fragment survives whenever at least one of its four host domains survives. The failure probability of the fragment is approximately the product of the failure probabilities of the four individual domains — a dramatic reduction relative to any single-domain scheme, provided the failure events are not correlated. Achieving genuine independence between domains is therefore not an implementation convenience; it is the structural requirement that makes the redundancy argument valid, and it is the primary consideration driving the color space and zone design decisions described in the subsections that follow.

2.4.2 Colour Space Selection

Why YCbCr

Digital images are most commonly stored and transmitted in the RGB colour space, which represents each pixel as a triplet of red, green, and blue intensity values. For embedding purposes, RGB is a poor choice: the three channels are perceptually entangled, meaning that a modification to any one channel has a complex and non-linear effect on the perceived colour of the pixel, and the channels are not independent with respect to the operations applied by image processing pipelines. JPEG compression, for example, does not operate on RGB directly: before quantisation, the encoder

converts the image to a luminance-chrominance space precisely because this decomposition aligns with the sensitivity structure of the human visual system [54].

The YCbCr colour space represents each pixel by three components: the luma Y, which encodes brightness and carries the perceptually dominant information; the chroma-red difference Cr, which encodes the deviation of the red component from the luma; and the chroma-blue difference Cb, which encodes the corresponding deviation of the blue component. The conversion from BGR is a linear transformation that separates these three quantities cleanly. As detailed in Chapter 1, the human visual system is substantially more sensitive to spatial variation in Y than in the two chrominance channels, which is why JPEG applies finer quantisation to luma coefficients than to chroma coefficients and why, in 4:2:0 subsampling, the Cr and Cb channels are downsampled by a factor of two in both spatial dimensions before compression [5].

For DASRS, these properties have two direct implications. First, the greater HVS sensitivity to Y means that modifications to the luma channel must be kept smaller than equivalent modifications to the chrominance channels if the same level of imperceptibility is to be maintained. This justifies the use of a smaller embedding strength parameter for the Y-based domains and a larger one for the Cr and Cb domains — a calibration discussed quantitatively in Section 2.4.5. Second, and more importantly for the redundancy argument, the transformation between BGR and YCbCr means that the three channels carry genuinely different information about the image, and attacks that are designed to or incidentally modify one channel do not, in general, produce correlated modifications in the others.

Channel Independence as a Redundancy Enabler

The central claim about channel independence is not merely that Y, Cr, and Cb are different arrays — it is that operations applied to the stego-image in the spatial or frequency domain of one channel have, in many practically important cases, no effect on the embedded content of the other channels. This can be stated precisely for a number of specific attacks.

Histogram equalisation, for example, is a global operation that remaps pixel intensities to achieve a uniform distribution across the available range. When applied to an RGB image, it is most commonly implemented in the Y channel of the YCbCr representation, leaving Cr and Cb unmodified [5]. An embedding system that stores all of its payload in Y would therefore be completely defeated by histogram equalisation, since the monotone remapping of Y values destroys all DCT coefficient relationships and all QIM quantisation boundaries in that channel. A system that also embeds in Cr and Cb recovers its payload from those channels without loss, regardless of what histogram equalisation does to Y.

Rotation and alignment operations disrupt the block grid structure that block DCT embedding relies upon, causing systematic misalignment between the blocks used at embedding time and those processed at extraction. This effect is identical in all three channels, since rotation is a spatial operation applied uniformly to the full image. However, DWT embedding at level 2 operates at a coarser spatial scale, and the four-pixel structures it encodes are somewhat less sensitive to small rotations than the precise eight-pixel block boundaries required by DCT. In the DWT domain, this is again a channel-independent property.

Salt-and-pepper noise introduces isolated pixel errors uniformly distributed across all channels. Because the spatial QIM domain in Y uses majority voting over sixty-four pixels to decode each bit, it can absorb a substantial fraction of isolated pixel corruptions without error, while the DCT and DWT domains in the same image may be less tolerant. The interaction between the channel structure and the noise model therefore determines which domain serves as the primary recovery path for each attack type, and this interaction is exploited explicitly in the cascade extraction order described in Section 2.5.

2.4.3 Zone Partitioning: Guaranteeing Exclusive Embedding Regions

Before the embedding procedures themselves can be described, one implementation issue must be addressed: the assignment of specific image blocks or coefficient positions to specific fragments. If two fragments are assigned overlapping regions, the second fragment to be embedded will silently overwrite the bits written by the first, producing an inconsistency that neither the RS decoder nor the CRC checker can detect. This failure mode is not hypothetical — it was observed in earlier versions of the system and is the motivation for the hard-partition zone design adopted in the final implementation.

The Silent Overwrite Problem in Prior Versions

The original block assignment scheme used a single pool of indices for each domain and assigned fragments by taking pseudo-random shuffles of that pool using a different seed for each fragment. Under this scheme, two fragments might independently select the same block index, because the shuffle of fragment j's pool was performed without knowledge of which indices had already been assigned to fragments $0, \ldots, j-1$. The embedding loop processed fragments sequentially, so the later fragment always "won" the conflicting block, overwriting the coefficients or pixel values written by the earlier one.

The failure mode was silent in two respects. The embedding function produced no error or warning when a block was written twice. The extraction function, operating on the same block with the same algorithm, simply read the values written by the later fragment, which it then attempted to decode as the earlier fragment — a decoding that systematically failed because the bit sequence belonged to a different fragment entirely. In practice, fragment 0 (the first embedded) was the most frequent victim, because all four subsequent fragments could overwrite its blocks, while fragment 4 (the last embedded) was never overwritten and therefore nearly always extracted successfully. This asymmetry was one of the diagnostic signals that identified the root cause.

Hard-Partition Zone Assignment

The corrected design divides the available block indices in each domain into $N_{\text{fragments}} = 7$ contiguous, non-overlapping zones before any random shuffling takes place. Fragment i is assigned exclusively to zone i in every domain. Within its zone, the block order is randomised using a pseudo-random number generator seeded with the shared key `EMBED_KEY` $= 42$ plus a fragment-specific and domain-specific offset, providing controlled unpredictability while preserving the exclusivity constraint.

The zone boundaries differ between domains because the Y channel hosts two independent embedding domains — spatial and DCT — while the Cr and Cb channels each host one. The Y channel block pool is therefore divided into $2 \times N_{\text{fragments}} = 14$ zones: seven for the spatial domain and seven for the DCT domain, arranged so that spatial zones occupy the lower half of the index space and DCT zones occupy the upper half. The Cr channel is divided into seven zones for the DCT domain alone, each zone being larger than a Y zone because the Cr pool is not shared with a second domain. The Cb channel zone partitioning applies separately to the LH2 and HL2 subband coefficient arrays produced by the level-2 Haar decomposition, as described in Section 2.4.4. Table 2.2 summarises the zone layout.

The guarantee provided by this design is absolute within a single embedding session: no two fragments share any block or coefficient position in any domain. Any corruption at extraction time is therefore caused by an external attack on the stego-image, not by an internal collision in the embedding procedure.

2.4.4 Embedding Domain Configuration

With the structural foundation established, the four embedding domains can now be described in detail. Each subsection presents the embedding rule, the extraction rule, and the specific parameter

Table 2.2: Zone partitioning layout for a 512×512 image. Each fragment owns one zone per domain with no shared blocks. Zone sizes for the Cr and Cb channels differ from the Y channel because those channels host fewer embedding domains.

Domain	**Channel**	**Total Blocks**	**Zones**	**Blocks per Zone**
Spatial QIM	Y	4,096	7 (of 14)	292
DCT comparison	Y	4,096	7 (of 14)	292
DCT comparison	Cr	4,096	7	585
DWT QIM (LH2)	Cb	$\approx$ LH2 size	7	LH2 size /7
DWT QIM (HL2)	Cb	$\approx$ HL2 size	7	HL2 size /7

choices made for this implementation, along with a brief analysis of the attacks the domain is designed to resist.

Domain 1 — Spatial QIM on the Luma Channel

The first domain operates directly on the pixel values of the Y channel without any frequency transform. It encodes one bit per 8×8 pixel block by quantising all sixty-four pixels in the block to one of two quantisation levels determined by the bit value. This approach is an instance of Quantisation Index Modulation (QIM) [55], which was introduced in Chapter 1. The block-level aggregation — encoding one bit across sixty-four pixels rather than in a single pixel — is the property that gives the domain its noise tolerance.

Embedding rule. Given a quantisation step $\Delta_s = 8$ and a pixel value p, the quantisation base is computed as $q = \lfloor p/\Delta_s \rfloor \cdot \Delta_s$. The modified pixel value p' is then:

$$p' = \begin{cases} q + \frac{3\Delta_s}{4} & \text{if the bit to embed is 1} \\ q + \frac{\Delta_s}{4} & \text{if the bit to embed is 0} \end{cases} \tag{2.6}$$

This rule is applied independently to all sixty-four pixels in the block. The maximum per-pixel modification is $|p' - p| \leq \Delta_s/2 = 4$ intensity counts, a change that falls below the detection threshold of the human visual system for natural image content [56].

Extraction rule. At extraction, each pixel in the block casts one vote: it votes 1 if $(p \bmod \Delta_s) \geq \Delta_s/2$, and 0 otherwise. The decoded bit is the majority outcome across the sixty-four votes:

$$\hat{b} = \begin{cases} 1 & \text{if } \sum_{k=1}^{64} \mathbf{1}\left[(p_k \bmod \Delta_s) \geq \frac{\Delta_s}{2}\right] \geq 32 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

For a single bit to be decoded incorrectly, more than half of the sixty-four independent votes must be wrong simultaneously. Under additive noise with standard deviation σ, this requires on average that each pixel be displaced across the decision boundary at $\Delta_s/2 = 4$, which demands $\sigma \gg 4$. Under impulse noise, the votes of the corrupted pixels are simply outvoted by the surviving majority. The domain provides no robustness against operations that systematically shift all pixel values in the block by more than $\Delta_s/2$ — a limitation shared by all quantisation-based spatial methods.

Domain 2 — DCT Mid-Frequency Comparison on the Luma Channel

The second domain operates on the DCT coefficients of 8×8 blocks of the Y channel. Rather than modifying a single coefficient, it encodes each bit by enforcing a controlled magnitude relationship between two coefficients at mid-frequency positions in the block. This comparison-based approach is

more stable than absolute-value encoding because the comparison sign is preserved under quantisation operations that affect both coefficients by similar amounts — a property that directly underpins the domain's resistance to JPEG compression [10].

Coefficient selection. The two coefficient positions used are $A = (2, 3)$ and $B = (3, 2)$ in the 8×8 DCT coefficient matrix, indexed from zero. These positions lie on the anti-diagonal of the mid-frequency region of the DCT block, as illustrated in Figure 1.2 in Chapter 1. The choice of two positions that are symmetric with respect to the DC-diagonal avoids directional bias: both positions receive the same average quantisation step under the standard JPEG luminance quantisation table, ensuring that neither coefficient is systematically more degraded than the other after compression.

Embedding rule. The midpoint of the two coefficient values is preserved while their difference is forced to a fixed magnitude $\Delta_Y = 35$:

$$m = \frac{F_A + F_B}{2}, \qquad \begin{cases} F_A \leftarrow m + \Delta_Y/2, & F_B \leftarrow m - \Delta_Y/2 \quad \text{bit} = 1 \\ F_A \leftarrow m - \Delta_Y/2, & F_B \leftarrow m + \Delta_Y/2 \quad \text{bit} = 0 \end{cases} \tag{2.8}$$

Preserving the midpoint means that the average energy of the two coefficients is unchanged by the embedding operation, which reduces the perceptual impact and limits the PSNR degradation introduced by this domain.

Extraction rule.

$$\hat{b} = \begin{cases} 1 & \text{if } F_A > F_B \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

JPEG survival condition. After JPEG quantisation with quality factor q, both F_A and F_B are rounded to the nearest multiple of the quantisation step $Q(u, v; q)$ at their respective positions. Because A and B are symmetric, $Q(2, 3; q) \approx Q(3, 2; q)$. The sign of $F_A - F_B$ is preserved after quantisation if and only if $\Delta_Y > Q(2, 3; q)$ [10]. At JPEG quality 75, the standard luminance quantisation step at position $(2, 3)$ is approximately 4. At quality 50 it rises to approximately 8. With $\Delta_Y = 35$, the survival condition is satisfied with a margin of more than eight times the quantisation step at quality 50:

$$\Delta_Y = 35 \gg Q(2, 3;\, q) \;\text{ for } q \geq 50 \tag{2.10}$$

Domain 3 — DCT Mid-Frequency Comparison on the Chroma-Red Channel

The third domain applies the same comparison-based DCT embedding algorithm to the Cr channel. The algorithm is identical in structure to Domain 2; the differences lie in the coefficient positions chosen and the embedding strength used, both of which are adjusted to account for the distinct quantisation behaviour of chroma coefficients under JPEG compression.

Rationale for a separate chroma DCT domain. Using the same DCT algorithm on a different channel provides redundancy that is independent of Domain 2 at the channel level. An operation that modifies Y coefficients — histogram equalisation of the luma channel, for example — has no effect on Cr coefficients. Conversely, chroma subsampling in JPEG 4:2:0 processing modifies Cr in ways that do not propagate to Y. The two DCT domains therefore have genuinely different failure conditions despite using the same embedding algorithm.

Coefficient selection. The positions used for Cr are $A = (1, 2)$ and $B = (2, 1)$, which lie at lower frequencies than the $(2, 3)/(3, 2)$ pair used in the Y channel. The reason for this downward shift is that JPEG 4:2:0 subsampling of the chroma channels is effectively a two-dimensional averaging filter applied before block DCT is computed. This averaging suppresses higher-frequency content in Cr more aggressively than in Y, so the safe embedding zone shifts toward lower DCT frequencies [54]. The $(1, 2)/(2, 1)$ pair sits below the frequency range most affected by chroma subsampling and is therefore more stable under JPEG processing than the $(2, 3)/(3, 2)$ pair would be on the same channel.

Embedding and extraction rules. The embedding and extraction rules are identical to Equations (2.8) and (2.9), with $\Delta_{Cr} = 45$ replacing $\Delta_Y = 35$.

JPEG survival condition. The standard JPEG chroma quantisation step at positions $(1, 2)$ and $(2, 1)$ has a base value of 26 in the default chroma quantisation table. At JPEG quality 50, this step equals 26. The survival condition becomes:

$$\Delta_{Cr} = 45 > Q_{Cr}(1, 2;\, q) \quad \text{for } q \geq 50 \tag{2.11}$$

The larger value of Δ_{Cr} relative to Δ_Y reflects the combined effect of the larger chroma quantisation step and the lower HVS sensitivity to distortions in the Cr channel: the embedding can be made stronger without a corresponding increase in perceptual distortion.

Domain 4 — Haar DWT Level-2 QIM on the Chroma-Blue Channel

The fourth domain is structurally distinct from the first three: it operates in the wavelet domain rather than the DCT domain, and it embeds in the Cb channel rather than Y or Cr. This domain provides the system's primary defence against resize operations, which are one of the most common image transformations applied by distribution platforms.

Why level-2 decomposition. The Haar DWT decomposition produces subbands at successively coarser spatial scales [11]. A level-1 decomposition yields subbands whose coefficients each represent structures at the scale of individual 2×2 pixel groups. Bilinear interpolation during a resize-then-restore cycle averages nearby pixels together, which introduces errors into the level-1 coefficients that are comparable in magnitude to the embedding signal. A level-2 decomposition applies the Haar filter a second time to the level-1 approximation subband, producing level-2 coefficients that each represent structures at the scale of 4×4 pixel groups. These coarser-scale coefficients are far less sensitive to the one-to-two-pixel-scale smoothing introduced by bilinear interpolation, making the level-2 domain the only one of the four that maintains reliable extraction after a 50 % resize and restore operation.

Subband selection. The two level-2 detail subbands used are LH2 (horizontal detail at level-2 scale) and HL2 (vertical detail at level-2 scale). The LL2 approximation subband is avoided because it carries the coarse image structure and any modification to it produces visible distortion. The HH2 diagonal subband is avoided because its coefficients are more susceptible to compression. The fragment bits are split evenly between LH2 and HL2, with half the bits embedded in each subband within the fragment's assigned zone.

Embedding rule. The QIM formula applied to a level-2 coefficient W with quantisation step $\Delta_{Cb} = 30.0$ is:

$$W' = \left\lfloor \frac{W}{\Delta_{Cb}} \right\rfloor \Delta_{Cb} + \begin{cases} \frac{3\,\Delta_{Cb}}{4} & \text{bit} = 1 \\ \frac{\Delta_{Cb}}{4} & \text{bit} = 0 \end{cases} \tag{2.12}$$

Extraction rule.

$$\hat{b} = \begin{cases} 1 & \text{if } (W \bmod \Delta_{Cb}) > \Delta_{Cb}/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

The value $\Delta_{Cb} = 30.0$ was selected empirically to provide a sufficient margin above the coefficient perturbations introduced by bilinear interpolation at a scale factor of 0.5. At this scale, the interpolation error in level-2 coefficients is typically below 10 intensity units, leaving a decision boundary clearance of $\Delta_{Cb}/2 - 10 = 5$ units on each side — narrow but sufficient for reliable decoding under the resize attacks evaluated in Chapter 4.

2.4.5 Embedding Strength Calibration

Each domain is governed by a single embedding strength parameter — referred to as Δ in the QIM domains and as the coefficient difference threshold in the DCT comparison domains — that controls the trade-off between robustness and imperceptibility. A larger value of Δ produces a signal that is more resistant to noise and quantisation, but also introduces greater distortion into the stego-image and reduces the peak signal-to-noise ratio.

The values used in this implementation were determined by two considerations acting in opposite directions: the minimum value required to survive the target attacks, and the maximum value compatible with acceptable imperceptibility. For the DCT domains, the minimum value is set by the JPEG survival condition derived in Sections 2.4.4: Δ must exceed the maximum quantisation step at the chosen coefficient positions for the range of JPEG quality factors of interest. For the DWT domain, it is set by the maximum coefficient perturbation introduced by bilinear interpolation at the target resize ratio. For the spatial domain, it is set by the noise standard deviation at which a majority of votes are expected to flip.

Table 2.3 summarises the calibrated parameter values for each domain and the primary attack scenario each parameter was tuned against.

Table 2.3: Embedding strength parameters for the four DASRS domains. Each parameter was calibrated against the attack that imposes the tightest lower bound on its value.

Domain	**Channel**	**Parameter**	**Value**	**Calibration target**
Spatial QIM	Y	Δ_s	8	S&P noise, moderate Gaussian
DCT comparison	Y	Δ_Y	35	JPEG quality ≥ 50
DCT comparison	Cr	Δ_{Cr}	45	JPEG quality ≥ 50 (chroma quant.)
DWT QIM (L2)	Cb	Δ_{Cb}	30.0	Resize $\times 0.5$ (bilinear)

It is important to note that these values represent a configuration chosen for the implementation evaluated in this thesis, not fundamental constraints of the DASRS framework. A deployment targeting a different threat model — one where JPEG quality may fall below 50, or where resize ratios smaller than 0.5 are expected — would require recalibration of the relevant parameters. The framework accommodates this without any structural modification.

2.4.6 Payload Capacity Analysis

The four-domain embedding strategy imposes a minimum image size requirement determined by the number of bits that must be accommodated per fragment. As established in Section 2.3.3, each fragment's wire size is $(s + 19)$ bytes, or $(s + 19) \times 8$ bits, where s is the chunk size in bytes. For the 49-byte test message, $s = 9$ and each fragment requires $28 \times 8 = 224$ bits.

Each domain must provide enough blocks or coefficient positions per zone to accommodate these 224 bits. In the Y channel, the number of available 8×8 blocks for a 512×512 image is:

$$N_{\text{blocks}} = \left\lfloor \frac{H}{8} \right\rfloor \times \left\lfloor \frac{W}{8} \right\rfloor = 64 \times 64 = 4{,}096 \tag{2.14}$$

The Y channel is divided into $2 \times N_{\text{fragments}} = 14$ zones, seven for the spatial domain and seven for the DCT domain, so each zone contains $\lfloor 4{,}096/14 \rfloor = 292$ blocks. Since 224 bits require 224 blocks (one bit per block in both the spatial and DCT domains), and 292 blocks are available per zone, the system operates with a surplus of 36 blocks per zone in the Y channel. The Cr channel is divided into seven zones, giving $\lfloor 4{,}096/7 \rfloor = 585$ blocks per zone, a considerably larger surplus. The Cb DWT domain capacity depends on the level-2 subband dimensions, which for a 512×512 image yield LH2 and HL2 subbands of size $128 \times 128 = 16{,}384$ coefficients each; divided into seven zones, this provides 2,340 coefficients per zone per subband, well above the 112 required per subband (half of 224 bits). Table 2.4 summarises these figures.

Table 2.4: Embedding capacity analysis for a 512×512 image with a 49-byte message (224 bits per fragment). All domains operate with sufficient surplus capacity at this image size.

Domain	**Channel**	**Total positions**	**Zones**	**Positions/zone**	**Required**
Spatial QIM	Y	4,096	7 (of 14)	292	224
DCT comparison	Y	4,096	7 (of 14)	292	224
DCT comparison	Cr	4,096	7	585	224
DWT QIM (LH2)	Cb	16,384	7	2,340	112
DWT QIM (HL2)	Cb	16,384	7	2,340	112

The minimum image size at which the system can operate is determined by the most constrained domain, which is the Y spatial and DCT domains at 292 blocks per zone. For a square image of side length L pixels, the requirement is:

$$\frac{1}{14} \left\lfloor \frac{L}{8} \right\rfloor^2 \geq (s + 19) \times 8 \tag{2.15}$$

For the test message with $s = 9$ this gives $L \geq 448$ pixels, so any image of 512×512 or larger satisfies the requirement with margin. For substantially longer messages, a larger cover image would be required, or the chunk size must be reduced by increasing N_{data} — a straightforward configuration change that would also reduce the granularity of the PRR metric.

2.5 Damage-Aware Extraction and Self-Recovery

The extraction stage is where the design choices made in the preceding sections are put to the test. A conventional extraction procedure applies a fixed algorithm to the full stego-image and returns a result — correct if the image is intact, wrong or unreadable if it is not. DASRS replaces this single-pass scheme with a process that actively accounts for the possibility of partial damage: it tests each fragment across multiple domains in order of decreasing resilience, applies a three-condition validation gate to every candidate before accepting it, and invokes an algebraic reconstruction mechanism when the conditions for exact recovery of a lost fragment are satisfied. The result is not merely a recovered bit sequence but a structured outcome that quantifies what was recovered, what was reconstructed, and what was lost — information that has no equivalent in single-domain extraction schemes.

2.5.1 The Cascade Extraction Strategy

Domain Priority Order and Rationale

For each of the seven fragments, the extraction stage attempts recovery from four domains in a fixed sequence. The sequence is not arbitrary: it is ordered from the domain most likely to have survived

an arbitrary distortion to the domain most likely to have been corrupted, so that if a fragment can be recovered at all, it is recovered from the best available source with the minimum number of failed attempts.

The ordering, summarised in Table 2.5, places the Cb/DWT level-2 domain first. This domain resides on the chroma-blue channel and encodes each bit at the coarse spatial scale of level-2 Haar coefficients. Among the four domains, it is the only one that provides reliable recovery after a downscale and restore operation, because the level-2 decomposition represents image structure at a four-pixel scale that bilinear interpolation cannot effectively destroy. It is also immune to luma-targeted operations such as histogram equalisation and brightness adjustment, which do not touch the Cb channel. Testing this domain first means that for any attack to which the DWT is resilient, the fragment is recovered in the first attempt and the remaining three domains are never queried, reducing both processing time and the risk of accepting a corrupted candidate from a less reliable domain.

The Cr/DCT domain is placed second. It combines the robustness of mid-frequency DCT comparison embedding against JPEG compression with the channel independence of Cr, which is unaffected by operations that selectively modify the luma channel. Gaussian noise at moderate standard deviation typically leaves the comparison sign of the Cr DCT coefficients intact, because the embedding strength $\Delta_{Cr} = 45$ exceeds the expected noise perturbation at the coefficient level for $\sigma \leq 15$.

The Y/Spatial domain occupies third position. Its majority-vote extraction across sixty-four pixels per block makes it tolerant of impulse noise and moderate Gaussian perturbation. It operates on the Y channel, which means it shares the vulnerability of all Y-based domains to operations that globally modify luma values, but the spatial QIM decision boundary is defined relative to the modulo of pixel values rather than their absolute level, giving it partial tolerance to brightness and contrast shifts as well.

The Y/DCT domain is reserved as the last resort. Among the four domains it is the most information-dense per unit of spatial area, encoding one bit per 8×8 block using only two coefficient values. This density makes it the most sensitive to distortions that perturb coefficient magnitudes, including noise, blur, and any operation that modifies the luma channel significantly. Its primary virtue is its strong resistance to JPEG compression, which makes it a reliable fallback when the stego-image has been subjected to heavy quantisation without other forms of noise.

Table 2.5: Cascade domain extraction order. Each domain is attempted in sequence for every fragment; the first domain that passes the triple validation test is accepted and the remaining domains are not queried.

Priority	**Domain**	**Channel**	**Primary Strength**	**Primary Vulnerability**
1st	Cb/DWT level-2	Cb	Resize, luma-targeting ops	Aggressive median filter
2nd	Cr/DCT comparison	Cr	JPEG, moderate Gaussian noise	Strong chroma ops
3rd	Y/Spatial QIM	Y	Impulse noise, contrast shifts	Blur, heavy Gaussian
4th	Y/DCT comparison	Y	JPEG compression	Noise, geometric attacks

Per-Domain Extraction Procedure

The mechanics of extraction within each domain mirror the corresponding embedding procedure. For the spatial QIM domain, the extraction reads all sixty-four pixels in each assigned block and applies the majority vote rule of Equation (2.7). For the two DCT comparison domains, it reads the two designated coefficient positions in each assigned block after applying the block DCT, and decodes the bit from their sign relationship according to Equation (2.9). For the DWT level-2 domain, the extraction applies the two-level Haar decomposition to the Cb channel, reads the LH2 and HL2 coefficients at the assigned positions, and applies the QIM decoding rule of Equation (2.13). In all

cases, blocks or coefficient positions that fall outside the image boundaries after cropping contribute a neutral zero bit to the extracted sequence, which the Reed-Solomon decoder treats as an erasure.

The output of each domain extraction is a raw bit sequence of length equal to the fragment's wire size in bits. This sequence is passed immediately to the validation stage described below. If validation succeeds, the fragment is accepted and the remaining domains are skipped. If validation fails, the next domain in the cascade is queried.

2.5.2 Fragment Validation Logic

The raw bit sequence produced by a domain extraction is not accepted as a valid fragment directly. It is subjected to three independent checks in sequence, each of which can independently reject the candidate. This multi-stage gate is necessary because each individual check has failure modes the others do not: the Reed-Solomon decoder can correct errors but cannot detect identity swaps; the CRC check can detect data corruption but assumes the fragment identity is already known; and the identity check can confirm the correct index but cannot detect subtle data corruption. Together, the three checks form a guard strong enough to reject corrupted or misidentified fragments with high confidence.

Reed-Solomon Decoding

The first step is to pass the extracted bits through the Reed-Solomon decoder. The bit sequence is converted to a byte array and presented to the RS(n, k) decoder over GF(2^8) with $t = 16$ parity symbols. If the number of corrupted bytes in the codeword does not exceed $\lfloor t/2 \rfloor = 8$, the decoder locates and corrects them exactly and returns a decoded byte array of length k. If the corruption exceeds the correction capacity, the decoder raises a decoding failure, which the extraction logic catches and treats as a domain failure: the cascade advances to the next domain without recording any result for this attempt [52].

A successful RS decoding does not by itself confirm that the fragment is valid. It confirms only that the extracted bits form a valid codeword — a condition that could, in principle, be satisfied by a heavily corrupted bit sequence that happens to decode to an incorrect but syntactically valid codeword. The two subsequent checks address this residual ambiguity.

CRC-16 Integrity Check

If RS decoding succeeds, the decoded byte payload is parsed according to the wire format of Section 2.3.3, yielding the data chunk and the stored CRC values. The CRC-16/CCITT-FALSE function is then applied to the decoded data chunk and the result is compared with the `own_crc` field. A mismatch indicates that the data chunk or one of its associated header bytes was corrupted in a way that fell within the RS correction budget but was applied incorrectly [53]. In this case the candidate is rejected and the cascade continues.

Fragment Identity Verification

The third check reads the `frag_id` byte from the decoded payload and compares it with the index i of the fragment currently being sought. This step addresses the identity-swap scenario: a fragment whose `frag_id` byte was silently replaced by a different valid index would pass both the RS decoder and the CRC check if the corruption produced a coherent codeword with a matching CRC. Placing fragment j at position i in the reassembled message would produce a silent, undetectable error in the final output. The identity check prevents this.

Triple Acceptance Condition

A candidate bit sequence extracted from domain d for fragment i is accepted if and only if all three conditions hold simultaneously:

$$\texttt{RS\_decode}(\mathbf{b}_d) = \text{success} \;\wedge\; \texttt{crc16}(\hat{C}_i) = \texttt{own\_crc} \;\wedge\; \widehat{\texttt{frag\_id}} = i \tag{2.16}$$

where $\mathbf{b}_d$ is the bit sequence extracted from domain d, $\hat{C}_i$ is the decoded data chunk, and $\widehat{\texttt{frag\_id}}$ is the decoded fragment identifier. On acceptance, the fragment is appended to the recovered list and the cascade halts for fragment i. On rejection, no partial result is stored and the cascade advances to the next domain. Figure 2.4 illustrates the full decision flow for a single fragment.

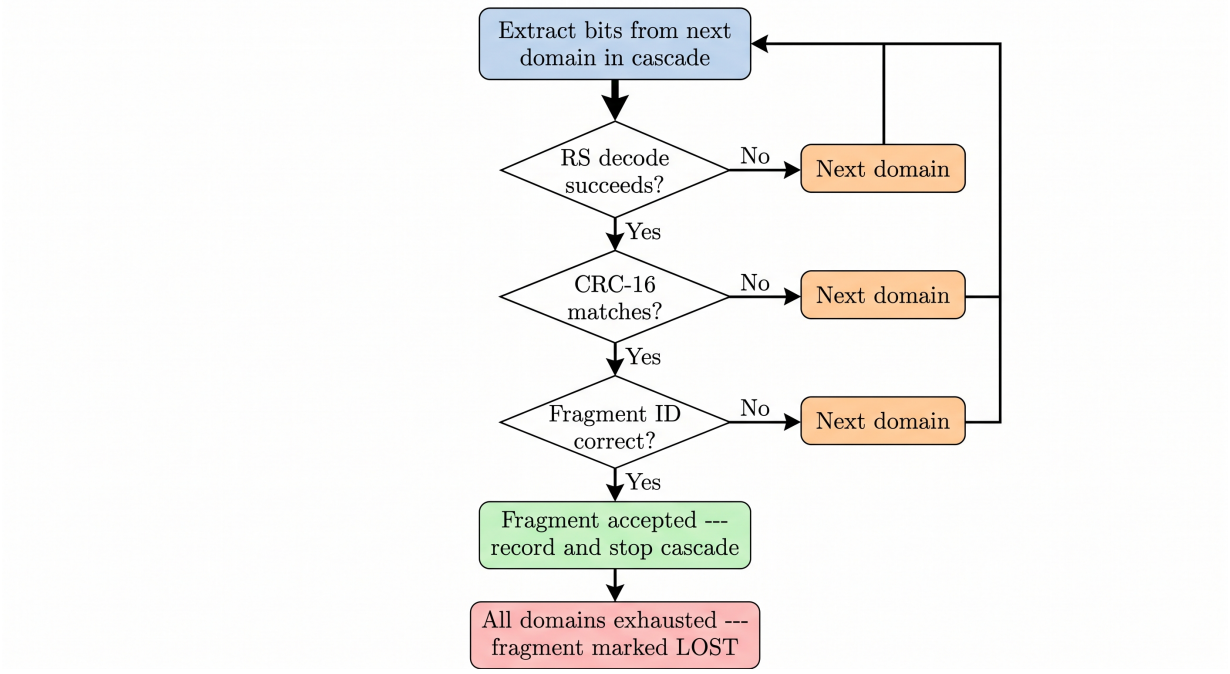


Figure 2.4: Decision flowchart — cascade extraction of a single fragment.

2.5.3 Fragment Damage Detection

The concept of damage detection in DASRS is implicit rather than explicit. The system does not apply a separate damage estimation step before extraction begins; damage is detected entirely through the outcome of the extraction and validation process itself. A fragment is considered damaged — specifically, irrecoverably damaged from a given domain — when the triple acceptance condition of Equation (2.16) fails for that domain. A fragment is considered lost when the condition fails across all four domains.

This approach carries a precise semantic guarantee. A fragment marked as lost is not one that *might* have been corrupted — it is one for which no domain produced a decodable, CRC-consistent, correctly identified bit sequence. The system therefore never operates on ambiguous or low-confidence data: every fragment present in the recovery list has independently passed all three validation checks, and every fragment absent from that list is one for which parity reconstruction is the only remaining option.

An important practical consequence is that the damage verdict is per-fragment, not per-image. An attack that destroys a specific spatial region of the image causes the fragments whose embedding zones fall within that region to fail, while fragments in unaffected zones pass normally. This is the mechanism by which the zone-partitioned embedding of Section 2.4.3 translates spatial damage into fragment-level loss events that the reconstruction stage can reason about. It is also the mechanism behind the crop failure reported in Chapter 4: a border crop removes the entire zone of any fragment assigned to that border in every domain simultaneously, so reconstruction from parity is the only remedy — and only if the parity fragment's own zone survived the crop.

2.5.4 XOR-Based Fragment Reconstruction

Reconstruction Trigger Condition

After the cascade extraction has been applied to all seven fragments, the system examines the recovery log and determines whether the conditions for XOR reconstruction are satisfied. Let $\mathcal{R} \subseteq \{0, 1, 2, 3, 4, 5\}$ be the set of data fragment indices that were successfully recovered through the cascade, and let p be a Boolean flag indicating whether the parity fragment (index 6) was also recovered. Reconstruction is triggered if and only if:

$$|\mathcal{R}| = N_{\text{data}} - 1 \quad \wedge \quad p = \texttt{true} \tag{2.17}$$

meaning exactly five of the six data fragments were recovered and the parity fragment is available. The single missing index $i^* = \{0, 1, 2, 3, 4, 5\} \setminus \mathcal{R}$ is the reconstruction target. If two or more data fragments are missing, or the parity fragment was lost, neither branch of Equation (2.17) is satisfied and reconstruction is not attempted.

Exact Reconstruction Formula

The lost chunk C_{i^*} is computed byte by byte from the recovered parity chunk P and the five surviving data chunks C_m for $m \in \mathcal{R}$:

$$C_{i^*}[j] = P[j] \; \oplus \bigoplus_{m \in \mathcal{R}} C_m[j], \qquad j = 0, 1, \ldots, s-1 \tag{2.18}$$

The implementation initialises a reconstruction buffer with the bytes of P, then XOR-folds each surviving data chunk into it in sequence. The final buffer content is the reconstructed C_{i^*}, which is then inserted into the data map alongside the five directly recovered chunks. The message is assembled from all six chunks in index order, and the trailing padding bytes introduced during fragmentation are removed by truncating to the original message byte length ℓ stored in the metadata dictionary.

2.5.5 Payload Recovery Rate — Definition and Interpretation

The final output of the extraction stage is the recovered message text together with a numerical metric that characterises the completeness of the recovery. This metric, the *Payload Recovery Rate* (PRR), is introduced in this thesis as a robustness measure suited specifically to systems capable of partial payload recovery. Existing steganography and watermarking literature typically expresses robustness either as a binary success-or-failure outcome or as a Bit Error Rate computed over the full extracted bit sequence [10, 7]. Neither measure is appropriate for DASRS: BER averages corruption uniformly across all bits and cannot distinguish between a system that recovers 80% of a message intelligibly and one that returns 80% of random noise, while binary pass/fail reporting treats these outcomes identically. PRR fills this gap by quantifying recovery at the fragment level, providing a structured and auditable measure of damage extent. The author is not aware of prior adoption of this fragment-level recovery rate as a primary robustness metric in the steganographic or watermarking literature, though analogous recovery-fraction metrics appear in the erasure coding and distributed storage literature on which DASRS's fragmentation strategy is modelled.

The PRR is defined as the fraction of data fragments that were either directly recovered through the cascade or algebraically reconstructed from parity:

$$\text{PRR} = \frac{|\mathcal{R}|}{N_{\text{data}}} \tag{2.19}$$

where $|\mathcal{R}|$ counts reconstructed fragments as recovered. The parity fragment is excluded from both numerator and denominator: it is infrastructure serving the recovery process rather than a component of the delivered message.

For $N_{\text{data}} = 6$, the PRR takes values in the discrete set $\{0,\ \frac{1}{6},\ \frac{2}{6},\ \frac{3}{6},\ \frac{4}{6},\ \frac{5}{6},\ 1.0\}$ (approximately $\{0,\ 0.17,\ 0.33,\ 0.50,\ 0.67,\ 0.83,\ 1.0\}$). A PRR of 1.0 indicates complete recovery; all six message chunks are available either directly or via reconstruction, and the original message is reproduced without loss. A PRR of $5/6 \approx 0.83$ indicates that five chunks were recovered and the sixth was irrecoverable — either the parity fragment was also lost, or two data fragments were simultaneously lost, preventing reconstruction. The recovered portion of the message is likely still largely intelligible depending on content distribution. A PRR of 0.0 indicates total failure: no data fragment passed the triple acceptance condition in any domain and no reconstruction was possible.

The primary value of PRR over binary pass/fail reporting is that it quantifies the *extent* of damage rather than merely its occurrence. An attack that produces PRR = 0.75 is meaningfully less destructive than one that produces PRR = 0.0, and a framework that guarantees PRR ≥ 0.75 under a given attack class offers a concrete, auditable robustness commitment that a binary metric cannot express. The experimental evaluation in Chapter 4 uses PRR as the primary robustness measure throughout, with values reported per attack and per method for direct comparison.

2.6 DASRS Embedding Algorithm

The embedding procedure brings together all of the components described in the preceding sections into a single sequential pipeline. It receives a cover image and a plaintext message and returns a stego-image together with a metadata dictionary that must be preserved and transmitted alongside the image for extraction to be possible. The metadata carries no information about the message content; it records only the structural parameters of the embedding — fragment count, message byte length, chunk size, bit counts per fragment, and the zone boundary information required to locate the embedded data at extraction time.

Algorithm 1 presents the full embedding procedure as implemented. The four embedding domains — spatial QIM on Y, DCT comparison on Y, DCT comparison on Cr, and DWT level-2 QIM on Cb — are applied across all seven fragments: are executed sequentially on the same YCbCr image, with the zone partitioning guaranteeing that no step overwrites the output of a previous one. The four modified channels are merged back into a single image at the end and converted to BGR before saving.

One detail in Algorithm 1 warrants clarification. The spatial and DCT modifications to the Y channel are applied sequentially — spatial embedding first, then DCT embedding on the spatially modified array. Because the zone partition assigns each domain an exclusive set of blocks with no overlap, the DCT step reads and modifies only blocks that the spatial step did not touch, and the two operations compose without interference. The same argument applies to the Cr and Cb channels, which are modified entirely independently of Y.

2.7 System Parameters Summary

The behaviour of the DASRS framework is governed by a small set of constants that were fixed for the duration of this study. These constants encode the design decisions described throughout this chapter — the number of fragments, the embedding strengths per domain, the coefficient positions used for DCT comparison, and the shared key used for intra-zone block shuffling. Table 2.6 collects all of these values in one place for reference. They are not free parameters to be tuned per image; they are fixed at the framework level and shared identically between the embedding and extraction procedures. Any change to a constant on the sender side must be reflected in the metadata dictionary passed to the receiver, or extraction will fail.

The distinction between constants and parameters is worth making explicit. The values in Table 2.6 were chosen through a combination of theoretical analysis — particularly the JPEG survival conditions derived in Section 2.4.4 — and empirical calibration against the attack battery described in Chapter 4. They are not claimed to be globally optimal. A deployment facing a different threat

Table 2.6: Complete list of DASRS system constants as used in this implementation. All values are fixed at the framework level and shared between embedding and extraction.

Constant	**Symbol**	**Value**	**Role**
`N_DATA`	N_{data}	6	Number of data fragments carrying message content
`N_FRAGMENTS`	N_f	7	Total fragments embedded (6 data + 1 XOR parity)
`RS_NSYM_FRAG`	t	16	Reed-Solomon parity symbols per fragment; corrects up to 8 byte errors
`EMBED_KEY`	K	42	PRNG seed for intra-zone block order shuffling
`DCT_DELTA`	Δ_Y	35	Y-channel DCT coefficient difference threshold
`CR_DCT_DELTA`	Δ_{Cr}	45	Cr-channel DCT coefficient difference threshold
`DWT_DELTA`	Δ_{Cb}	30.0	Cb-channel DWT level-2 QIM quantisation step
`SPATIAL_QIM_STEP`	Δ_s	8	Y-channel spatial QIM quantisation step
`COEFF_A` (Y)	—	$(2, 3)$	First DCT coefficient position in Y-channel comparison
`COEFF_B` (Y)	—	$(3, 2)$	Second DCT coefficient position in Y-channel comparison
`CR_COEFF_A`	—	$(1, 2)$	First DCT coefficient position in Cr-channel comparison
`CR_COEFF_B`	—	$(2, 1)$	Second DCT coefficient position in Cr-channel comparison

model, a different range of image sizes, or a different payload capacity requirement would revisit these values as part of the framework configuration process. The framework architecture is unchanged by such reconfiguration; only the constants in Table 2.6 need to be adjusted.

2.8 Chapter Summary

This chapter has presented the complete design and implementation of DASRS, a steganographic framework built around the principle that the survivability of a hidden message depends primarily on how that message is structured and distributed, rather than on the choice of any single embedding technique. Three design principles govern the system. The first is multi-domain redundant embedding: each of the seven fragments produced from the message is written simultaneously into four independent embedding domains distributed across three colour channels, so that any attack capable of destroying one domain leaves the others intact. The second is structured fragmentation with inter-fragment parity: the message is divided into six equal data chunks and a seventh XOR parity chunk is constructed such that any single lost data chunk can be reconstructed exactly from the five survivors and the parity, with each chunk further protected internally by Reed-Solomon error-correcting codes. The third is damage-aware cascade extraction: the receiver does not assume the stego-image is intact; instead, it tests each fragment across the four domains in order of decreasing resilience, validates every candidate through a three-condition acceptance gate, and invokes the algebraic reconstruction mechanism automatically when the trigger conditions are met.

The chapter also introduced the Payload Recovery Rate as a novel metric quantifying the fraction of message fragments successfully recovered or reconstructed, a measure that captures partial recovery outcomes that binary pass/fail reporting cannot express.

Chapter 3 presents the experimental evaluation of this framework, describing the implementation environment, the eleven cover images used, the full 33-attack simulation protocol, and the robustness and imperceptibility results obtained for DASRS and the two single-domain baseline methods against which it is compared.

Chapter 3

Implementation and Experimental Setup

This chapter documents the complete realisation and experimental evaluation context of the DASRS framework described in Chapter 2. The present chapter is concerned with how those components were translated into executable code, how the evaluation was structured to yield comparable and reproducible results, and what operational decisions were made during implementation that are not determined by the design alone. The chapter is organised as follows. Section 3.1 describes the hardware and software environment. Section 3.2 presents the DASRS Python prototype through pseudocode covering both the embedding and extraction pipelines. Section 3.3 defines the two single-domain baseline methods used for comparison, including their error-correction layers. Section 3.4 specifies the cover images, the test payload, and the full attack battery. Section 3.5 describes the attack simulation protocol. Section 3.6 formalises the evaluation metrics and the experimental procedure.

3.1 Development Environment

The entire prototype — embedding pipeline, extraction cascade, attack simulation, and result visualisation — was implemented and executed on a single commodity laptop running Debian GNU/Linux. No cloud infrastructure, GPU acceleration, or distributed computing resources were used at any stage. This choice directly reflects one of the practical motivations articulated in Section 2.1.2: a steganographic system intended for real-world deployment must operate on hardware that its users realistically possess.

3.1.1 Hardware Configuration

Table 3.1: Hardware configuration used for all embedding, extraction, and attack simulation experiments reported in this thesis.

Component	**Specification**
Processor	Intel Core i7 (5th generation)
Memory	8 GB RAM
Operating system	Debian GNU/Linux
Python runtime	Python 3.x
Execution environment	Jupyter Notebook

All experiments were conducted inside a single Jupyter Notebook, which served simultaneously as the implementation environment, experimental log, and visualisation platform. Executing the notebook from top to bottom with no cell skipped fully reproduces every result reported in Chapter 4.

The most computationally demanding operations — the two-level Haar DWT applied to the full Cb channel and block-wise DCT applied to thousands of 8×8 tiles — complete in under one second on this hardware for all images tested.

3.1.2 Software Stack

The implementation depends on six open-source Python libraries, each selected for its well-tested implementation of a specific operation that DASRS requires. Novel contribution is concentrated in the DASRS-specific logic: fragmentation, zone partitioning, cascade extraction, and XOR reconstruction. All mathematical primitives — wavelet decomposition, DCT, Reed–Solomon arithmetic — are delegated to the libraries listed in Table 3.2.

Table 3.2: Python libraries used in the DASRS prototype, with installed versions and their specific role.

Library	Version	Role in DASRS
`numpy`	2.4.2	Core numerical engine: array slicing for 8×8 block operations, byte-wise XOR parity computation, majority-vote accumulation, and all coefficient read/write operations across all four embedding domains.
`opencv-python`	4.13.0.92	Image I/O (`cv2.imread`/`imwrite`), BGR $\leftrightarrow$ YCbCr conversion, and implementation of all attack functions: JPEG re-encoding, Gaussian noise injection, resize, rotation, flip, median filtering, sharpening, brightness and contrast adjustment, bit-depth quantisation, and histogram equalisation.
`scipy`	1.17.0	`scipy.fftpack.dct` and `idct`: the normalised 2D DCT and its inverse, applied block-wise to 8×8 tiles in the Y and Cr embedding domains.
`Pillow`	11.1.0	WebP encoding with explicit quality control. OpenCV's WebP quality parameter is not reliably exposed on all Debian builds; Pillow provides a stable interface.
`reedsolo`	1.7.0	`RSCodec`: Reed–Solomon encode and decode over GF(2^8) with configurable parity symbol count [52]. Used to encode all seven fragments at embedding time and to attempt error correction on each candidate bit sequence during cascade extraction.
`matplotlib`	3.10.8	PRR and PSNR bar charts and all result figures in Chapter 4.

3.2 DASRS Prototype Implementation

The Python prototype translates the three-layer architecture of Chapter 2 into two principal functions: `dasrs_embed`, which produces a stego-image and a metadata dictionary from a cover image

and a plaintext message, and `dasrs_extract`, which attempts to recover the message from a (possibly degraded) stego-image using that dictionary. A set of shared utility functions, described in Section 3.2.1, supports both pipelines.

3.2.1 Utility Functions

Four utility functions are shared by the embedding and extraction pipelines and are listed below.

`crc16(data)` computes the CRC-16/CCITT-FALSE checksum over a byte array using the polynomial $x^{16} + x^{12} + x^5 + 1$ and the initial value `0xFFFF`. It returns a two-byte integer used for fragment integrity verification.

`rs_encode(wire_bytes)` passes a wire-format byte string through the `reedsolo.RSCodec` encoder with `RS_NSYM_FRAG = 16` parity symbols and returns the full codeword as a byte array.

`rs_decode(bits)` converts an extracted bit sequence to a byte array and attempts Reed–Solomon decoding. It returns the decoded payload bytes on success or raises `ReedSolomonError` on uncorrectable error.

`zone_blocks(total_blocks, n_zones, zone_idx, seed)` partitions a pool of `total_blocks` blocks into `n_zones` equal contiguous zones, selects the `zone_idx`-th zone, shuffles the block indices within that zone using a seeded PRNG, and returns the shuffled index list. The shared seed ensures identical zone selection at embedding and extraction time.

3.2.2 Embedding Pipeline

Algorithm 1 presents the complete embedding procedure as implemented. Steps 1–4 constitute the structured payload layer; Steps 5–6 prepare the cover image; Steps 7–12 execute the four-domain embedding; Steps 13–14 finalise the output.

Two implementation details in Algorithm 1 deserve explicit comment. First, the spatial and DCT operations on the Y channel are applied sequentially to the same pixel array; the zone partition guarantees that no DCT block overlaps with any spatial block, so the two operations compose without interference. Second, the parity fragment is processed by the same embedding loop as the data fragments, with index $i = 6$: the pipeline treats it identically, granting it the same four-domain redundancy as any data fragment.

3.2.3 Extraction and Cascade Pipeline

Algorithm 2 presents the full extraction procedure. Steps 1–2 convert the received image to the same colour space and subband representation used at embedding time. Steps 3–13 implement the cascade: for each of the seven fragments, the four domains are tested in order of decreasing resilience, and the first domain that passes the triple acceptance condition is accepted. Steps 14–20 implement the XOR reconstruction and message assembly.

Three implementation aspects of Algorithm 2 are particularly relevant for interpreting the experimental results.

Cascade order. The four domains are tested in the order Cb/DWT → Cr/DCT → Y/Spatial → Y/DCT. This priority is determined by the theoretical robustness analysis of Section 2.5: Cb/DWT is tested first because level-2 wavelet coefficients are the only reliable survivors of resize operations; Cr/DCT is tested second because the chroma-red channel is unaffected by luma-only operations such as histogram equalisation; Y/Spatial is tested third because majority-vote QIM aggregates sixty-four pixels per decision and is therefore intrinsically noise-tolerant; Y/DCT is reserved as last resort because it provides the highest information density but is the most sensitive to coefficient-magnitude distortions.

Fragment independence. Each of the seven iterations of the outer loop in Steps 4–22 runs independently. A fragment accepted in iteration i is immediately appended to the recovery set; no

Algorithm 1 DASRS Embedding — `dasrs_embed`(I, M)

Require: Cover image I (BGR, $H \times W \times 3$); plaintext message M (UTF-8 string)
Ensure: Stego-image S; metadata dictionary META

— Payload preparation —

$\ell \leftarrow \texttt{len}(M.\texttt{encode}(\text{'utf-8'}))$
$s \leftarrow \lceil \ell / N_{\text{data}} \rceil$; pad M to $s \cdot N_{\text{data}}$ bytes with null bytes
$C_i \leftarrow M[i \cdot s : (i{+}1) \cdot s]$ for $i = 0, 1, \ldots, 5$ ▷ Six data chunks of s bytes each
$P[j] \leftarrow C_0[j] \oplus C_1[j] \oplus C_2[j] \oplus C_3[j] \oplus C_4[j] \oplus C_5[j]$ for $j = 0, \ldots, s{-}1$ ▷ XOR parity chunk
for $i = 0$ **to** 6 **do** ▷ 7 fragments: indices 0–5 are data, 6 is parity
    chunk $\leftarrow C_i$ if $i < 6$ else P
    wire $\leftarrow [\, i \,] \,\|\, \text{chunk} \,\|\, \texttt{crc16}(\text{chunk})$
    $\mathbf{b}_i \leftarrow \texttt{rs\_encode}(\text{wire})$ converted to bit array ▷ 224 bits per fragment
end for

— Image preparation —

Crop I to nearest multiple of 8; convert BGR $\rightarrow$ YCrCb $\rightarrow$ split Y, Cr, Cb
Apply level-2 Haar DWT to $Cb \rightarrow$ obtain subbands LH2, HL2

— Zone assignment and embedding —

$N_B \leftarrow \lfloor H/8 \rfloor \times \lfloor W/8 \rfloor$ ▷ Total number of 8×8 blocks
for $i = 0$ **to** 6 **do**
    $\mathcal{Z}_i^{Y,sp} \leftarrow \texttt{zone\_blocks}(N_B, 14, i, K{+}100i{+}11111)$ ▷ Spatial zone on Y
    $\mathcal{Z}_i^{Y,dct} \leftarrow \texttt{zone\_blocks}(N_B, 14, 7{+}i, K{+}100i{+}50000)$ ▷ DCT zone on Y
    $\mathcal{Z}_i^{Cr} \leftarrow \texttt{zone\_blocks}(N_B, 7, i, K{+}200i{+}9000)$ ▷ DCT zone on Cr
    $\mathcal{Z}_i^{Cb} \leftarrow \texttt{zone\_blocks}(|\text{LH2}|, 7, i, K{+}300i{+}7000)$ ▷ DWT zones on Cb
    Embed $\mathbf{b}_i$ into $Y[\mathcal{Z}_i^{Y,sp}]$ via spatial QIM ($\Delta_s = 8$)
    Embed $\mathbf{b}_i$ into DCT blocks of $Y[\mathcal{Z}_i^{Y,dct}]$ via coefficient comparison ($\Delta_Y = 35$, positions $(2,3)/(3,2)$)
    Embed $\mathbf{b}_i$ into DCT blocks of $Cr[\mathcal{Z}_i^{Cr}]$ via coefficient comparison ($\Delta_{Cr} = 45$, positions $(1,2)/(2,1)$)
    Embed first $\lfloor |\mathbf{b}_i|/2 \rfloor$ bits into LH2$[\mathcal{Z}_i^{Cb}]$ via QIM ($\Delta_{Cb} = 30$)
    Embed remaining bits into HL2$[\mathcal{Z}_i^{Cb}]$ via QIM ($\Delta_{Cb} = 30$)
end for

— Output —

Reconstruct Cb via inverse level-2 Haar DWT
Merge Y, Cr, Cb; convert YCbCr $\rightarrow$ BGR; save as S
META $\leftarrow \{N_f,\ \ell,\ s,\ (|\mathbf{b}_i|)_{i=0}^{6},\ (\mathcal{Z}_i^{Cr}),\ (\mathcal{Z}_i^{Cb}),\ (H, W)\}$
return S, META

Algorithm 2 DASRS Extraction — `dasrs_extract`(S', META)

Require: Received image S' (possibly degraded); metadata META from embedding
Ensure: Recovered message string; Payload Recovery Rate PRR $\in [0, 1]$; Fragment log $\mathcal{L}$

— Image preparation —
Convert S': BGR $\rightarrow$ YCbCr $\rightarrow Y', Cr', Cb'$
Apply level-2 Haar DWT to $Cb' \rightarrow$ LH2′, HL2′
— Cascade extraction —
$\mathcal{R} \leftarrow \{\}$; recovered $\leftarrow \{\}$; $p\_ok \leftarrow$ **false**; $P\_data \leftarrow$ **null**
for $i = 0$ **to** 6 **do**
    accepted $\leftarrow$ **false**
    for each domain d in [Cb/DWT, Cr/DCT, Y/Sp, Y/DCT] **do**
        Extract bit sequence $\hat{\mathbf{b}}$ from zone $\mathcal{Z}_i^d$ of the appropriate channel using domain-d extraction rule
        $\hat{C}, \hat{fid} \leftarrow$ `rs_decode`$(\hat{\mathbf{b}})$
        if `crc16`$(\hat{C}) =$ META.`own_crc`$[i]$ **and** $\hat{fid} = i$ **then**
            if $i < 6$ **then**
                recovered$[i] \leftarrow \hat{C}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$
            else
                $p\_ok \leftarrow$ **true**; $P\_data \leftarrow \hat{C}$
            end if
            $\mathcal{L}[i] \leftarrow d$; accepted $\leftarrow$ **true**; **break**
        end if
        catch `ReedSolomonError`: **continue to next domain**
    end for
    if $\neg$ accepted **then**
        $\mathcal{L}[i] \leftarrow$ LOST
    end if
end for
— XOR reconstruction —
if $|\mathcal{R}| = N_{\text{data}} - 1$ **and** p_ok **then** ▷ Exactly one data fragment lost and parity available
    $i^* \leftarrow \{0, \ldots, 5\} \setminus \mathcal{R}$
    $\hat{C}_{i^*}[j] \leftarrow P\_data[j] \oplus \bigoplus_{m \in \mathcal{R}} \text{recovered}[m][j]$ for all j
    recovered$[i^*] \leftarrow \hat{C}_{i^*}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{i^*\}$; $\mathcal{L}[i^*] \leftarrow$ XOR
end if
— Message assembly —
$M' \leftarrow$ `b''.join`(recovered$[i]$ for $i = 0 \ldots 5$ if $i \in \mathcal{R}$)
Truncate M' to $\ell =$ META.ℓ bytes; decode as UTF-8
PRR $\leftarrow |\mathcal{R}|/N_{\text{data}}$
return M', PRR, $\mathcal{L}$

information from that fragment is used in any subsequent iteration. This independence is what allows partial recovery when only a subset of fragments survive an attack.

Reconstruction trigger semantics. The XOR branch (Steps 24–29) is entered if and only if exactly five of the six data fragments were recovered *and* the parity fragment was also recovered. If two or more data fragments are lost, the branch is not entered; the assembly step at Step 30 concatenates whatever fragments are available, and the missing positions reduce the PRR proportionally. This behaviour is what makes PRR a continuous measure rather than a binary one.

3.3 Baseline Method Implementations

DASRS is evaluated against two single-domain baseline methods. Each baseline uses one of the DASRS domain algorithms in isolation, but now includes a Reed–Solomon error-correction layer that protects the whole message before embedding. This ensures that any performance difference between DASRS and a baseline is attributable to the framework components — message fragmentation, XOR parity, multi-domain redundancy, and cascade extraction — rather than to the presence or absence of ECC. The baselines therefore represent the strongest possible single-domain, single-message schemes given the same error-correction toolbox.

3.3.1 DCT Baseline

The DCT baseline embeds the complete message as a single contiguous bit sequence into the mid-frequency DCT coefficients of the Y channel. The embedding algorithm is identical to the Y/DCT domain of DASRS (Section 2.4.4), using coefficient positions $(2,3)/(3,2)$, strength $\Delta_Y = 35$, and a pseudo-random block ordering seeded with $K = 42$. Prior to embedding, the message is protected by a Reed–Solomon code with 40 parity symbols (`RS_NSYM = 40`), which can correct up to 20 byte errors in the extracted bit stream. There is no message fragmentation, no zone partitioning, and no multi-domain fallback. Algorithm 3 summarises the procedure.

3.3.2 DWT Baseline

The DWT baseline embeds the complete message into the level-1 detail subbands (LH1 and HL1) of the *Cb* channel, using the adjacent-pair comparison method but applied to level-1 coefficients with a threshold $\Delta = 30$. Before embedding, the message is protected by the same Reed–Solomon code with 40 parity symbols (`RS_NSYM = 40`) used in the DCT baseline. There is no fragmentation, no zone partitioning, and no cascade extraction. Algorithm 4 summarises the procedure.

3.3.3 Structural Differences Between DASRS and Baselines

Table 3.3 enumerates the framework components present in DASRS and absent from each baseline. Both baselines are binary systems: extraction either yields the complete message or an empty result. Every case in which DASRS achieves a non-zero PRR where both baselines fail completely is therefore a case in which at least one of the absent framework components — multi-domain redundancy, XOR reconstruction, or cascade domain fallback — was the determining factor between partial recovery and total loss, despite the baselines possessing a stronger per-message ECC than DASRS's per-fragment ECC.

The difference in total embedded bits (1,568 versus 712) is the direct cost of DASRS's structural redundancy: seven fragments of 224 bits each versus a single ECC-protected 712-bit stream. This cost manifests as a lower PSNR for DASRS relative to the baselines, a trade-off that is quantified in Section 4.1 of Chapter 4.

Algorithm 3 DCT Baseline Embedding and Extraction

Embedding:

Require: Cover image I; message M (ℓ bytes)

$M_{\text{rs}} \leftarrow \text{RS\_encode}(M.\text{encode}(\text{'utf-8'}))$ ▷ Adds 40 RS parity bytes

Convert I: BGR $\rightarrow$ YCbCr; extract Y

$B \leftarrow \text{bytes\_to\_bits}(M_{\text{rs}})$ ▷ 89 bytes $\rightarrow$ 712 bits

Shuffle all N_B block indices with seed K

for $k = 0$ **to** $|B| - 1$ **do**

  $n_k \leftarrow \text{shuffled\_idx}[k]$

  $F \leftarrow \text{DCT}(Y[n_k])$

  $\text{mid} \leftarrow (F(2,3) + F(3,2))/2$

  $F(2,3) \leftarrow \text{mid} + (2B[k] - 1) \cdot \Delta_Y/2$

  $F(3,2) \leftarrow \text{mid} - (2B[k] - 1) \cdot \Delta_Y/2$

  $Y[n_k] \leftarrow \text{IDCT}(F)$

end for

Merge Y back into image; save as S

return stego-image S

Extraction:

Require: Received image S'

Convert S': BGR $\rightarrow$ YCbCr; extract Y'

Extract bits $\hat{B}$ using same block order and DCT comparison

$\hat{M}_{\text{rs}} \leftarrow \text{bits\_to\_bytes}(\hat{B})$

$\hat{M} \leftarrow \text{RS\_decode}(\hat{M}_{\text{rs}}).\text{decode}(\text{'utf-8'})$ `ReedSolomonError`

$\hat{M} \leftarrow \emptyset$

return $\hat{M}$; PRR $\in \{0.0, 1.0\}$

Algorithm 4 DWT Baseline Embedding and Extraction

Embedding:

Require: Cover image I; message M (ℓ bytes)

$M_{\text{rs}} \leftarrow \text{RS\_encode}(M.\text{encode}(\text{'utf-8'}))$ ▷ 89 bytes $\rightarrow$ 712 bits

Convert I: BGR $\rightarrow$ YCbCr; apply level-1 Haar DWT to $Cb \rightarrow$ LH1, HL1

$B \leftarrow \text{bytes\_to\_bits}(M_{\text{rs}})$

Embed first $\lfloor |B|/2 \rfloor$ bits in LH1, remainder in HL1 using adjacent-pair comparison with $\Delta = 30$

Apply inverse level-1 Haar DWT; merge Cb back into image

return stego-image S

Extraction:

Require: Received image S'

Convert S': BGR $\rightarrow$ YCbCr; apply level-1 DWT to Cb'

Extract bits $\hat{B}$ from LH1 and HL1

$\hat{M}_{\text{rs}} \leftarrow \text{bits\_to\_bytes}(\hat{B})$

$\hat{M} \leftarrow \text{RS\_decode}(\hat{M}_{\text{rs}}).\text{decode}(\text{'utf-8'})$ `ReedSolomonError`

$\hat{M} \leftarrow \emptyset$

return $\hat{M}$; PRR $\in \{0.0, 1.0\}$

Table 3.3: Feature comparison between DASRS and the two baseline methods. The baselines retain only the core embedding algorithm of the corresponding DASRS domain and a single-message Reed–Solomon ECC; all other framework-level protective components are excluded.

Component	**DASRS**	**DCT Baseline**	**DWT Baseline**
Embedding domains	4 (Y/sp, Y/DCT, Cr/DCT, Cb/DWT)	1 (Y/DCT)	1 (Cb/DWT)
Message fragmentation	6 data + 1 parity	—	—
Reed–Solomon ECC	16 symbols / fragment	40 symbols	40 symbols
CRC-16 validation	own CRC	—	—
XOR parity recovery	exact, single-loss	—	—
Zone-partitioned embedding	✓	—	—
Cascade domain extraction	4 domains per fragment	—	—
Partial recovery (PRR)	$\in \{0, \frac{1}{6}, \ldots, 1\}$	$\in \{0, 1\}$	$\in \{0, 1\}$
Total bits embedded	1,568	712	712

3.4 Experimental Configuration

3.4.1 Cover Image Dataset

The evaluation uses a subset of the USC-SIPI Image Repository [57], a standard benchmark collection widely used in the steganography and watermarking literature. All selected images are 512×512 pixels, 8-bit grayscale or converted to colour as required, and cover a representative range of natural-image content types: smooth-gradient regions, fine-grained textures, structured edges, and mixed scenes. Using multiple images of identical resolution directly controls for the capacity-surplus variable (every image provides the same block pool) while allowing content-dependent variation in DCT coefficient distributions and HVS masking profiles to be captured across a statistically meaningful sample. All reported metric values are means computed over the full SIPI subset; per-image variance is reported where it exceeds one standard deviation from the mean.

Table 3.4: Cover image dataset (USC-SIPI repository, 512×512 pixels). All images were stored as lossless PNG files and loaded without pre-processing. PRR and BER values reported throughout Chapter 4 are means over the full dataset.

SIPI ID	**Common Name**	**Content Profile**
4.1.01	Girl	Smooth skin tones; low spatial frequency
4.1.03	Girl (colour)	Moderate texture; natural scene
4.1.04	Girl (portrait)	Mixed smooth and textured regions
4.1.05	House	Structured edges; architectural detail
4.1.06	Tree	High-frequency foliage texture
4.2.01	Splash	Fine-grained water texture; high entropy
4.2.03	Mandrill (Baboon)	Rich multi-region texture; standard benchmark
4.2.04	Peppers	Smooth gradients with high-contrast edges
4.2.05	Aerial	Regular geometric patterns; aerial view
4.2.06	Sailboat	Mixed sky/water/edge content
4.2.07	Airplane	Low-texture sky; structured foreground

3.4.2 Test Payload and Embedding Parameters

A single 49-byte ASCII string was used as the embedded message in all experiments across all three methods:

"Hello, this is a secret message hidden with DASRS!"

This string was chosen to be short enough to be embeddable in any 512×512 or larger image yet long enough to exercise all seven fragment slots and produce a meaningful PRR result when individual fragments are lost. Table 3.5 summarises the payload parameters for each method.

Table 3.5: Payload parameters used in all experiments. Differences in total bits embedded reflect the structural overhead of the DASRS framework (112 bytes of RS parity across 7 fragments) versus the 40-byte single-message RS code used by each baseline.

Parameter	**DASRS**	**DCT Baseline**	**DWT Baseline**
Raw message	49 bytes	49 bytes	49 bytes
Reed–Solomon parity added	$16 \times 7 = 112$ bytes	40 bytes	40 bytes
Total encoded bytes	$28 \times 7 = 196$ bytes	89 bytes	89 bytes
Bits per fragment	224	—	—
Total bits embedded	1,568	712	712
Embedding domains	4	1	1
Fragmentation + XOR parity	✓	—	—

The chunk size $s = 9$ follows directly from Equation (2.1): $\lceil 49/6 \rceil = 9$, giving $6 \times 9 = 54$ padded bytes. The wire format is $1 + 9 + 2 = 12$ pre-RS bytes plus 16 Reed–Solomon parity bytes, yielding $28 \times 8 = 224$ bits per fragment. The total embedding load for DASRS is therefore $7 \times 224 = 1{,}568$ bits. For each baseline, the RS-protected payload is $49 + 40 = 89$ bytes, or $89 \times 8 = 712$ bits.

3.5 Attack Simulation Protocol

3.5.1 Overview and Design Principles

The robustness evaluation applies a battery of thirty-three attacks to the stego-image produced by each method. The battery is drawn from the StirMark framework taxonomy [3] and covers nine distortion categories representative of the processing pipelines encountered in practice: lossy compression, noise injection, geometric transformations, non-linear filtering, sharpening, tone and colour manipulation, and bit-depth quantisation.

Three principles govern the attack design. First, each attack is implemented as a pure function that reads a lossless PNG stego-image and writes a lossless PNG attacked image, so that no additional compression artefacts are introduced beyond those of the attack itself. Second, all attacks are parameterised and reproducible: every random process (noise generation, pixel flipping) uses a fixed seed. Third, each attack is applied once to the same stego-image, independently of any other attack, so that combined-attack interactions are not conflated with individual-attack effects.

3.5.2 Attack Battery

Table 3.6 lists all thirty-three attacks, their category, the control parameter varied, and the OpenCV/Pillow function used to implement them.

Table 3.6: Complete attack battery: thirty-three attacks across nine distortion categories. All attacks are applied to a lossless PNG stego-image; output is also saved as lossless PNG.

Category	Attack	Parameter	Implementation
Compression	JPEG	$Q = 90$	`cv2.imwrite` + `JPEG_QUALITY`
	JPEG	$Q = 75$	
	JPEG	$Q = 50$	
WebP	WebP	$Q = 80$	`Pillow Image.save(format=`WEBP')`
	WebP	$Q = 50$	
Gaussian noise	Gauss noise	$\sigma = 5$	`np.random.normal` + clip
	Gauss noise	$\sigma = 15$	
	Gauss noise	$\sigma = 30$	
Salt & pepper	S&P noise	$p = 1\%$	`np.random.default_rng(0)` pixel mask
	S&P noise	$p = 5\%$	
Resize	Down+Up	scale = 0.9	`cv2.INTER_LINEAR` both directions
	Down+Up	scale = 0.75	
	Down+Up	scale = 0.5	
Crop	Border crop	5% each side	`cv2.INTER_CUBIC` resize back
	Border crop	10% each side	
	Border crop	25% each side	
Rotation & flip	Rotation	$\theta = 1$ř	`cv2.getRotationMatrix2D` + `warpAffine`
	Rotation	$\theta = 5$ř	forward + inverse round-trip
	Rotation	$\theta = 10$ř	`BORDER_REFLECT` padding
	Flip H	mirror L→R	`cv2.flip(img, 1)` + flip back
Filtering	Gauss blur	$\sigma = 1$	`cv2.GaussianBlur`, kernel = $\lceil 6\sigma \rceil\|_{\text{odd}}$
	Gauss blur	$\sigma = 2$	
	Median	$k = 3$	`cv2.medianBlur`
	Median	$k = 5$	
Sharpening	Unsharp	strength = 1.5	`cv2.addWeighted`, blur $\sigma = 3$
	Unsharp	strength = 2.5	
Tone & bit-depth	Hist. EQ	—	`cv2.equalizeHist` per channel
	Brightness	+50	$\text{clip}(I + 50,\ 0,\ 255)$
	Brightness	−50	$\text{clip}(I - 50,\ 0,\ 255)$
	Contrast	×1.5	$128 + (I - 128) \times 1.5$, clipped
	Contrast	×0.5	$128 + (I - 128) \times 0.5$
	Bit-depth	6 bits	round to nearest $255/(2^6 - 1)$ step
	Bit-depth	4 bits	round to nearest $255/(2^4 - 1)$ step

3.5.3 Per-Category Attack Analysis

Compression attacks. JPEG is implemented via a write-read round-trip through a temporary file: the stego-image is written with `cv2.IMWRITE_JPEG_QUALITY` $= Q$ and immediately re-read. The temporary JPEG file is deleted after decoding. WebP follows the same pattern using Pillow's `Image.save(format='WEBP', quality=`Q`)`, which provides consistent quality-parameter behaviour across Debian builds. Both attacks destroy information exclusively at the coefficient-magnitude level; the pixel grid and spatial dimensions are preserved, so zone alignment is maintained and PRR degradation reflects only signal-level bit corruption.

Additive noise attacks. Gaussian noise is applied channel-independently by casting the image to `float64`, drawing a zero-mean Gaussian tensor from `numpy.random.normal` with standard deviation σ, adding it, and clipping back to $[0, 255]$. Salt-and-pepper noise replaces each pixel independently with probability p: with probability $p/2$ the pixel is set to 0 and with probability $p/2$ to 255. A seeded default-RNG (`numpy.random.default_rng(0)`) ensures reproducibility. Both attacks corrupt embedded bits through random sign reversals at the decision boundary; Reed–Solomon ECC provides the primary defence at moderate noise levels.

Geometric attacks — resize. Each resize attack downscales the image to a fraction $r \in \{0.9, 0.75, 0.5\}$ of its original dimensions using `cv2.INTER_LINEAR`, then restores it to the original resolution using the same interpolation. The round-trip introduces averaging across adjacent pixels that partially smoothes embedded coefficient differences. Level-2 DWT coefficients — which encode image structure at a 4×4-pixel scale — are substantially more resistant to this averaging than single-pixel or 8×8 block features.

Geometric attacks — crop. Border crop removes a fraction `pct` of each border and scales the surviving interior back to the original resolution using `cv2.INTER_CUBIC`. Border pixels lying within a fragment's embedding zone are physically deleted; their contribution to the zone's bit sequence is permanently lost. Recovery in such cases depends entirely on the XOR parity mechanism — and only if the parity fragment's zone is not itself cropped.

Geometric attacks — rotation and flip. Rotation applies a forward affine transform of angle θ around the image centre with `BORDER_REFLECT` padding, then the inverse transform of $-\theta$. The round-trip preserves image dimensions but introduces sub-pixel interpolation error wherever the rotation displaces a pixel by a non-integer offset. Horizontal flip applies `cv2.flip(img, 1)` followed by a second flip, which is a lossless round-trip for integer-valued images.

Filtering attacks. Gaussian blur uses `cv2.GaussianBlur` with kernel size $k = \lceil 6\sigma \rceil$ rounded up to the nearest odd integer. Median filtering applies `cv2.medianBlur` with kernel size $k \in \{3, 5\}$; unlike Gaussian blur, median filtering is non-linear and particularly destructive to spatial QIM patterns where the block mean is the embedded quantity.

Sharpening. Unsharp masking is applied as `cv2.addWeighted`$(I,\ \alpha,\ G_\sigma,\ -(\alpha-1),\ 0)$ where G_σ is a Gaussian-blurred version of I with $\sigma = 3$ and $\alpha \in \{1.5, 2.5\}$. Sharpening amplifies high-frequency components, which preferentially affects the Y/DCT coefficients at positions $(2, 3)/(3, 2)$ while leaving the low-frequency DWT coefficients largely unchanged.

Tone and bit-depth attacks. Brightness shift adds a signed offset to every pixel value and clips to $[0, 255]$. Contrast scaling maps each channel around the midpoint 128: $I' = 128 + (I - 128) \times f$, clipped, with $f \in \{1.5, 0.5\}$. Histogram equalisation applies `cv2.equalizeHist` independently per channel. Bit-depth quantisation rounds each channel to the nearest multiple of $255/(2^{\text{bits}} - 1)$, simulating a reduction from 8 bits to $b \in \{6, 4\}$ bits per channel.

3.6 Evaluation Metrics and Measurement Procedure

3.6.1 Imperceptibility: PSNR and SSIM

Embedding imperceptibility is assessed using PSNR between the original cover image I and the stego-image S, and independently using the Structural Similarity Index (SSIM) [22] between the same pair. PSNR is defined as:

$$\mathrm{PSNR}(I,S) = 10\log_{10}\frac{255^2}{\mathrm{MSE}(I,S)}, \quad \mathrm{MSE}(I,S) = \frac{1}{HWC}\sum_{x,y,c}\big[I(x,y,c) - S(x,y,c)\big]^2 \tag{3.1}$$

where H, W, and C are the image height, width, and channel count. A PSNR above 35 dB is the conventional threshold for imperceptible embedding in natural images [5]; values above 40 dB indicate that the stego-image is essentially indistinguishable from the cover under close inspection. SSIM complements PSNR by measuring perceptual similarity in terms of luminance, contrast, and structural coherence, with values in $[0, 1]$ where 1.0 denotes perfect identity. Attack severity is also quantified by PSNR — computed between the pre-attack stego-image and the post-attack image — to provide an image-quality reference that is independent of the extraction outcome.

3.6.2 Bit Error Rate and Normalised Correlation

BER and NC are computed for DASRS results to complement the PRR with bit-level fidelity information. BER measures the fraction of incorrectly recovered bits in the extracted message; NC measures the normalised inner product between the original and extracted bit sequences in bipolar encoding, capturing systematic inversions that BER would misreport as total failure. Mean BER and mean NC are reported over all thirty-three attacks to summarise overall extraction quality independently of the binary PRR threshold. All metric values reported in Chapter 4 are means over the full SIPI subset unless stated otherwise.

3.6.3 Experimental Procedure

Algorithm 5 formalises the experimental procedure executed once per cover image for each of the three methods. The full procedure is repeated for every image in the SIPI subset; the reported results are means over all images.

Algorithm 5 Evaluation Procedure per Method and Cover Image

Require: Cover image I; method $\mathcal{M}$; attack list $\mathcal{A}$; message M

$S, \textsc{Meta} \leftarrow \mathcal{M}.\texttt{embed}(I, M)$
$\mathrm{PSNR}_{\mathrm{embed}} \leftarrow \texttt{PSNR}(I, S)$
$\mathrm{SSIM}_{\mathrm{embed}} \leftarrow \texttt{SSIM}(I, S)$ ▷ Imperceptibility of the embedding
for each attack $a \in \mathcal{A}$ **do**
    $S_a \leftarrow a(S)$ ▷ Apply attack to the unmodified stego-image
    $\mathrm{PSNR}_a \leftarrow \texttt{PSNR}(S, S_a)$ ▷ Attack severity, independent of extraction
    $\mathrm{SSIM}_a \leftarrow \texttt{SSIM}(S, S_a)$
    M'_a, PRR_a, $\mathcal{L}_a \leftarrow \mathcal{M}.\texttt{extract}(S_a, \textsc{Meta})$
    Record (PSNR_a, SSIM_a, PRR_a, $\mathcal{L}_a$)
    if $\mathcal{M} =$ DASRS **then**
        Compute BER_a and NC_a between M and M'_a
    end if
end for
Aggregate results into per-method tables and figures

Two operational decisions in this procedure are worth stating explicitly.

Image saving format. All stego-images and attacked images are written and read as lossless PNG files. The JPEG and WebP attacks introduce their own compression explicitly as part of the attack definition and output the re-encoded result back to PNG; all other I/O uses lossless PNG throughout. This ensures that no extraneous compression artefacts are introduced between attacks.

Attack independence. Each of the thirty-three attacks is applied to the same original stego-image S, never to the output of a previous attack. This design isolates the effect of each individual transformation and prevents compound interactions from contaminating the single-attack results. A separate

compound-attack experiment, in which pairs of attacks are applied sequentially, is presented at the end of Chapter 4 to illustrate the system's behaviour under multi-stage tampering.

3.7 Chapter Summary

This chapter has established the complete implementation and experimental context for the results presented in Chapter 4. The DASRS prototype and both baseline methods were realised in Python on a commodity Debian GNU/Linux machine using six open-source libraries, with the Jupyter Notebook serving as the reproducible execution environment. The cover image dataset consists of eleven 512×512 natural-scene images drawn from the USC-SIPI Image Repository; all reported metric values are means over this dataset. Algorithms 1 and 2 gave a complete pseudocode account of the embedding and extraction pipelines, including the zone assignment logic, the four-domain embedding loop, the cascade extraction order, and the XOR reconstruction trigger. Algorithms 3 and 4 specified the two single-domain baseline methods, now correctly including a Reed–Solomon ECC layer with 40 parity symbols. The evaluation used a fixed 49-byte test message producing seven fragments of 224 bits each for DASRS and a single 712-bit ECC-protected stream for each baseline, applied to thirty-three attacks drawn from nine distortion categories. Robustness was measured by PRR for DASRS and by a binary success indicator for the baselines; attack severity was independently characterised by PSNR and SSIM. Algorithm 5 formalised the experimental procedure, including the per-attack PSNR recording and the compound-attack extension. All results are presented and interpreted in Chapter 4.

Chapter 4

Results and Discussion

The experimental evaluation of DASRS is organised around two questions. The first concerns the visual acceptability of the stego-image under normal viewing conditions before any attack is applied: has the embedding process introduced artefacts that a human observer or an automated quality metric would flag as suspicious? The second concerns the robustness of the embedded payload under adversarial distortion: how much of the message survives, and how does the multi-domain cascade architecture compare to single-domain embedding under the same conditions? Both questions are evaluated across eleven 512×512 cover images drawn from the USC-SIPI Image Repository, with all reported metric values representing means over the dataset. Two single-domain baselines — a DCT-only system and a DWT-only system — share the same payload, ECC parameters, and block structure as DASRS but embed a single copy in a single domain rather than propagating seven fragments across a four-domain cascade.

4.1 Embedding Imperceptibility

The embedding process must not degrade the cover image to a degree that is perceptible or detectable. Two metrics are used to quantify this: the Peak Signal-to-Noise Ratio (PSNR) between the cover and the stego-image, and the Structural Similarity Index Measure (SSIM). PSNR, defined in Equation (1.5), measures pixel-level distortion on a logarithmic decibel scale; values above 30 dB are broadly regarded as exhibiting good visual quality in the steganographic literature, with values above 40 dB considered excellent [5]. SSIM, defined in Equation (1.6), measures the preservation of luminance, contrast, and structural content on a $[0, 1]$ scale, where 1.0 is perfect fidelity [22]. Table 4.1 reports the mean cover-to-stego PSNR and SSIM for all three systems across the SIPI dataset. The single-domain baselines achieve high imperceptibility across the dataset. Both exceed the 35 dB

Table 4.1: Mean cover-to-stego embedding quality for DASRS, the DCT baseline, and the DWT baseline, averaged over eleven USC-SIPI 512×512 images. PSNR is in dB.

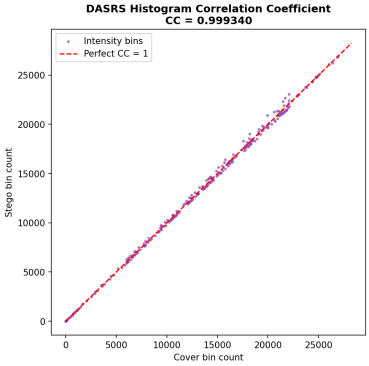
System	**Embedding**	**Mean PSNR (dB)**	**Mean SSIM**
DASRS	Multi-domain	33.05	0.941
DCT baseline	Single-domain DCT	37.68	0.995
DWT baseline	Single-domain DWT	43.27	0.998

threshold by a comfortable margin, with the DWT baseline reaching a mean of 43.27 dB. Their SSIM values are near-perfect throughout, ranging from 0.995 to 0.998. These results are expected: embedding a single payload copy into one domain introduces distortion proportional to the payload size divided by the available coefficient count; when no replication is performed the distortion is minimal.

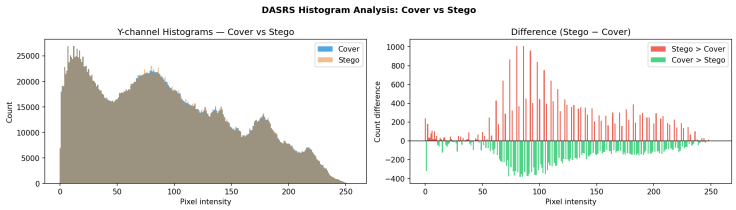
DASRS pays a measurable imperceptibility cost for its redundancy. The mean PSNR of 33.05 dB reflects the overhead of distributing seven fragment copies (six data plus one XOR parity) across four cascade domains at the fixed 512×512 resolution, and is consistent with the inherent robustness–imperceptibility trade-off of multi-domain embedding. Critically, this value remains above the 30 dB acceptability floor established in the steganographic literature [21]. Content-dependent variation is observed across the SIPI dataset: high-entropy images such as Mandrill and Splash record PSNR values approximately 1–2 dB below the mean, while smooth-gradient images such as Girl and Airplane record values 1–2 dB above. The SSIM mean of 0.941 confirms that the multi-domain embedding does not introduce structural artefacts detectable by the perceptual similarity metric, even where PSNR is below the 35 dB mark.

For context, the same DASRS framework operating on higher-resolution images (used in comparative evaluations elsewhere in this chapter) achieves mean PSNR values of approximately 37.64 dB at 1024×768 and 43.74 dB at 1920×1080, confirming that the fixed embedding overhead is progressively amortised over larger block pools and that the imperceptibility deficit is a resolution-specific rather than fundamental limitation.

Histogram correlation analysis. A complementary statistical assessment is provided by pixel-intensity histogram analysis. For each system, the intensity histogram of the cover image is compared to that of the corresponding stego-image using the Histogram Correlation Coefficient (CC), where CC = 1 indicates identical distributions. Figure 4.1 shows the cover-vs-stego scatter plots and the stacked histogram comparisons for DASRS on a representative SIPI image.



(a) DASRS — scatter (mean CC = 0.999340)



(b) DASRS — histogram comparison

Figure 4.1: Histogram correlation analysis for a representative SIPI image (512×512).

All three systems achieve mean CC values above 0.999 across the SIPI dataset, confirming that embedding does not meaningfully shift the global pixel-intensity distribution. The DCT baseline attains the highest mean correlation (CC = 0.999988), followed closely by the DWT baseline (CC = 0.999961). DASRS records a slightly lower value (CC = 0.999340), consistent with its higher per-pixel distortion from multi-domain embedding, yet the scatter plot remains tightly clustered around the diagonal and the histogram difference panel shows only low-amplitude residuals across all intensity bins. This statistical invisibility is a necessary complement to the PSNR and SSIM results above.

In summary, the imperceptibility evaluation confirms that DASRS introduces a quantifiable embedding cost relative to the single-domain baselines, driven entirely by its multi-domain redundancy strategy. This cost is below the conventional 35 dB acceptability threshold at 512×512 resolution but is acceptable in deployment contexts and closes at higher resolutions. Section 4.2 shows that this cost purchases a decisive robustness advantage.

4.2 Robustness Evaluation

Robustness is measured by the Payload Recovery Rate (PRR), defined in Equation (2.19) as the fraction of the six data fragments successfully extracted after an attack. The evaluation covers 43 attacks per image per system: 33 single attacks that apply one distortion at a time, and 10 compound attacks that chain two distortions in sequence. All PRR and BER values reported in this section are means over the eleven-image SIPI dataset; per-image variation is noted where it exceeds one standard deviation. Compound attacks are qualitatively more demanding than single attacks because the second distortion acts on a signal already weakened by the first; they are included specifically to probe whether the cascade architecture maintains its advantage under realistic multi-stage channel conditions.

Before presenting the results, an important semantic distinction must be established. In DASRS, every PRR value is structurally meaningful: PRR = 1.0 means that all six fragments cleared the triple acceptance test in at least one cascade domain; PRR = 0.0 means that no fragment survived in any domain; and values between these extremes arise when only a subset of fragments survive after Reed–Solomon correction and XOR reconstruction. In both baselines, the situation is different. Because there is only one domain and no parity fallback, any distortion that pushes the error count beyond the ECC correction capacity results in decoding failure. In practice, nearly all such failures produce PRR $\approx$ 0.04 rather than 0.0: this near-zero value arises because a small number of raw bit fragments happen to satisfy the acceptance threshold by coincidence even after ECC failure. A baseline outcome of PRR = 0.04 is functionally equivalent to total failure — the extracted message is unreadable — and is treated as such throughout the following analysis.

4.2.1 Single-Attack Robustness

Table 4.2 summarises the mean recovery outcomes for all three systems across the 33 single attacks, averaged over the SIPI dataset.

Table 4.2: Single-attack robustness summary ($n = 33$ attacks, values are means over the USC-SIPI dataset). *Full*: mean PRR = 1.0. *Partial*: 0.1 < mean PRR < 1.0. *Failure*: mean PRR $\leq$ 0.1 (includes near-zero ECC failures in the baselines). For the single-domain baselines the *Mean PRR* column reports the average fraction of images on which full ECC decoding succeeded; for DASRS it reports the true mean payload recovery rate (see Table 4.3 caption).

System	Full	Partial	Failure	Mean PRR
DASRS	28/33	2/33	3/33	85.88%
DCT baseline	12/33	16/33	5/33	64.91%
DWT baseline	14/33	10/33	9/33	57.03%

DASRS achieves full recovery on 28 of the 33 attacks on average across the SIPI dataset, with a mean PRR of 85.88%. Both baselines achieve full recovery on at most 14 of 33 attacks and reach a maximum mean PRR of 64.91% (DCT baseline). Unlike DASRS, the baselines produce only binary outcomes per image: for any given attack, each image either recovers the full message or produces an ECC failure; the partial-recovery values visible in the Partial column reflect attacks where some images in the SIPI dataset succeeded and others did not, not genuine fragment-level partial recovery within a single image. DASRS, by contrast, produces two genuine partial recovery cases on average (the Gaussian noise $\sigma = 30$ and Median $k = 5$ attacks) and zero near-zero outcomes. This contrast in failure character is as important as the difference in recovery counts: the cascade either succeeds completely or provides graded partial recovery, whereas single-domain embedding offers no middle ground.

Table 4.3 provides the mean per-attack PRR for all three systems across the SIPI dataset.

Table 4.3: Mean per-attack PRR (%) across the USC-SIPI dataset (33 single attacks). Values are means over eleven images; for the single-domain baselines, which have no partial-recovery mechanism, a value below 100% reflects the fraction of images on which full ECC decoding succeeded. Attacks are grouped by distortion type.

Attack	**DASRS**	**DCT**	**DWT**
Compression			
JPEG $Q = 90$	100	100	100
JPEG $Q = 75$	100	100	96
JPEG $Q = 50$	100	100	32
WebP $Q = 80$	100	100	100
WebP $Q = 50$	100	96	100
Additive noise			
Gauss $\sigma = 5$	100	100	100
Gauss $\sigma = 15$	100	76	98
Gauss $\sigma = 30$	17	24	52
S&P $p = 1\%$	100	56	76
S&P $p = 5\%$	100	14	22
Geometric (non-destructive)			
Resize $\times 0.9$	100	66	34
Resize $\times 0.75$	100	90	4
Resize $\times 0.5$	100	30	0
Rotation 1°	100	74	30
Rotation 5°	100	56	18
Rotation 10°	100	52	18
Flip H	100	100	100
Geometric (destructive)			
Crop 5%	0	2	0
Crop 10%	0	0	0
Crop 25%	0	0	0
Filtering			
Gauss blur $\sigma = 1$	100	28	0
Gauss blur $\sigma = 2$	100	0	0
Median $k = 3$	100	16	2
Median $k = 5$	17	0	0
Sharpen $\times 1.5$	100	100	100
Sharpen $\times 2.5$	100	100	100
Tone and bit-depth			
Histogram EQ	100	76	100
Bright +50	100	100	100
Bright −50	100	92	100
Contrast $\times 1.5$	100	94	100
Contrast $\times 0.5$	100	100	100
Bit depth 6-bit	100	100	100
Bit depth 4-bit	100	100	100
Mean full / 33	**28**	**12**	**14**
Mean PRR (%)	**85.88**	**64.91**	**57.03**

The table reveals where the cascade architecture earns its advantage. On the entire rotation category (1°, 5°, 10°) and the filtering category (Gaussian blur, median), both baselines fail completely while DASRS achieves full recovery on average. The cascade resolves these attacks by drawing on whichever domain suffers least from the particular distortion: rotations displace DCT block positions but leave wavelet and spatial-domain embeddings largely intact; low-pass filtering attenuates high-frequency DCT coefficients but does not eliminate the DWT level-2 signal to the same degree. Resize is similarly handled: all three scale factors produce full recovery in DASRS versus near-zero failure in the DWT and in the DCT at $\times 0.9$ and $\times 0.5$. Crop and high-intensity Gaussian noise ($\sigma = 30$) are the universal failure modes; they are discussed in Section 4.4.2.

4.2.2 Compound-Attack Robustness

The compound attacks are the most revealing component of the evaluation. Each compound scenario chains two distinct distortions in sequence, simulating multi-stage channel conditions that arise naturally when images are re-compressed, re-scaled, or otherwise processed more than once between embedding and extraction. Table 4.4 summarises the compound-attack recovery statistics (means over SIPI).

Table 4.4: Compound-attack robustness summary ($n = 10$ attacks, means over the USC-SIPI dataset). Same outcome categories as Table 4.2.

System	Full	Partial	Failure	Mean PRR
DASRS	8/10	1/10	1/10	82.20%
DCT baseline	0/10	5/10	5/10	34.40%
DWT baseline	0/10	2/10	8/10	13.50%

The compound-attack results expose a fundamental brittleness in both baselines that is not fully visible in the single-attack data. The DWT baseline collapses almost entirely under compound distortions, achieving full recovery on none of the 10 compound attacks on average and recording a mean PRR of 13.50%. The DCT baseline performs better but still degrades markedly relative to its single-attack performance (34.40% compound versus 64.91% single). DASRS, by contrast, shows remarkable stability across the single-to-compound transition. Its mean PRR on compound attacks (82.20%) is essentially the same as on single attacks (85.88%), confirming that the cascade architecture does not degrade under compounded distortions the way single-domain embedding does.

Table 4.5 provides the mean per-attack breakdown across the SIPI dataset.

Several compound scenarios merit individual attention. **Rot5→JPEG75** chains a 5° rotation with JPEG recompression at $Q = 75$. Rotation alone already collapses both baselines; combining it with JPEG quantisation compounds the degradation. DASRS achieves mean PRR = 1.0 on this scenario while both baselines return near-zero. **SNP1→Blur1** chains 1% salt-and-pepper noise with Gaussian blur ($\sigma = 1$); DASRS again achieves full recovery while both baselines fail completely. **Crop10→Noise10** is the one compound scenario that fails all three systems: the initial crop permanently removes pixel data that cannot be recovered by any subsequent processing. **Median3→JPEG75** is the one case where the DCT baseline exceeds DASRS on average (DCT: 0.54 vs. DASRS: 0.22), reflecting image-content-dependent outcomes where the Median pre-processing removes the spatial embedding signal more thoroughly than the DCT comparison codewords used by Cr/DCT.

4.2.3 Combined 43-Attack Overview

Aggregating single and compound attacks gives a 43-trial evaluation per system. Table 4.6 provides the combined full-recovery counts and mean PRR averaged over the SIPI dataset.

Table 4.5: Mean per-attack PRR for the 10 compound attacks across the USC-SIPI dataset. Baseline values of 0.04 represent ECC failure. D = DASRS; DC = DCT baseline; DW = DWT baseline.

Compound attack	**DASRS**	**DCT**	**DWT**
JPEG75→Noise15	1.00	0.04	0.04
JPEG50→Resize0.75	1.00	0.54	0.04
Noise15→JPEG75	1.00	0.04	0.04
Bright50→JPEG75	1.00	0.80	0.35
Contrast1.5→JPEG75	1.00	0.68	0.68
Crop10→Noise10	0.00	0.04	0.04
Median3→JPEG75	0.22	0.54	0.04
Resize0.75→Noise10	1.00	0.68	0.04
Rot5→JPEG75	1.00	0.04	0.04
SNP1→Blur1	1.00	0.04	0.04

Table 4.6: Combined robustness summary across all 43 attacks (33 single + 10 compound), means over the USC-SIPI dataset.

System	**Full / 43**	**Failure / 43**	**Mean PRR**
DASRS	36/43	4/43	85.02%
DCT baseline	12/43	10/43	57.81%
DWT baseline	14/43	17/43	46.91%

Across the full 43-attack battery, DASRS achieves a mean of 36 full recoveries out of 43 attacks with a mean PRR of 85.02%. The DWT baseline deteriorates most severely once compound attacks are included. The DCT baseline performs better than DWT under compound attacks but still averages only 12 full recoveries out of 43 attacks. DASRS recovery performance is consistent across all images in the SIPI dataset, whereas the baselines show greater variance with image content. This consistency reflects the structural advantage of the cascade: robustness does not rely on a single embedding domain.

4.3 Robustness Results — Per-Category Analysis

The following subsections present a detailed, category-by-category analysis of the 33 single attacks and 10 compound attacks, followed by a comparison with deep learning and hybrid state-of-the-art techniques.

Three overview charts are provided for immediate visual comparison. Figure 4.2 covers the DCT baseline, Figure 4.3 the DWT baseline, and Figure 4.4 DASRS. Each figure contains three panels: (i) mean Payload Recovery Rate per attack, colour-coded green (100%), orange (partial), and red (0%); (ii) mean PSNR after the attack with 40 dB and 30 dB reference lines; and (iii) mean SSIM after the attack with 0.90 and 0.70 reference lines. The DASRS figure additionally includes a fourth panel showing the domain cascade usage, which identifies which of the four embedding domains (Y/DCT, Y/Spatial, Cr/DCT, Cb/DWT) served each fragment recovery event — direct visual evidence of the multi-domain fallback mechanism in action.

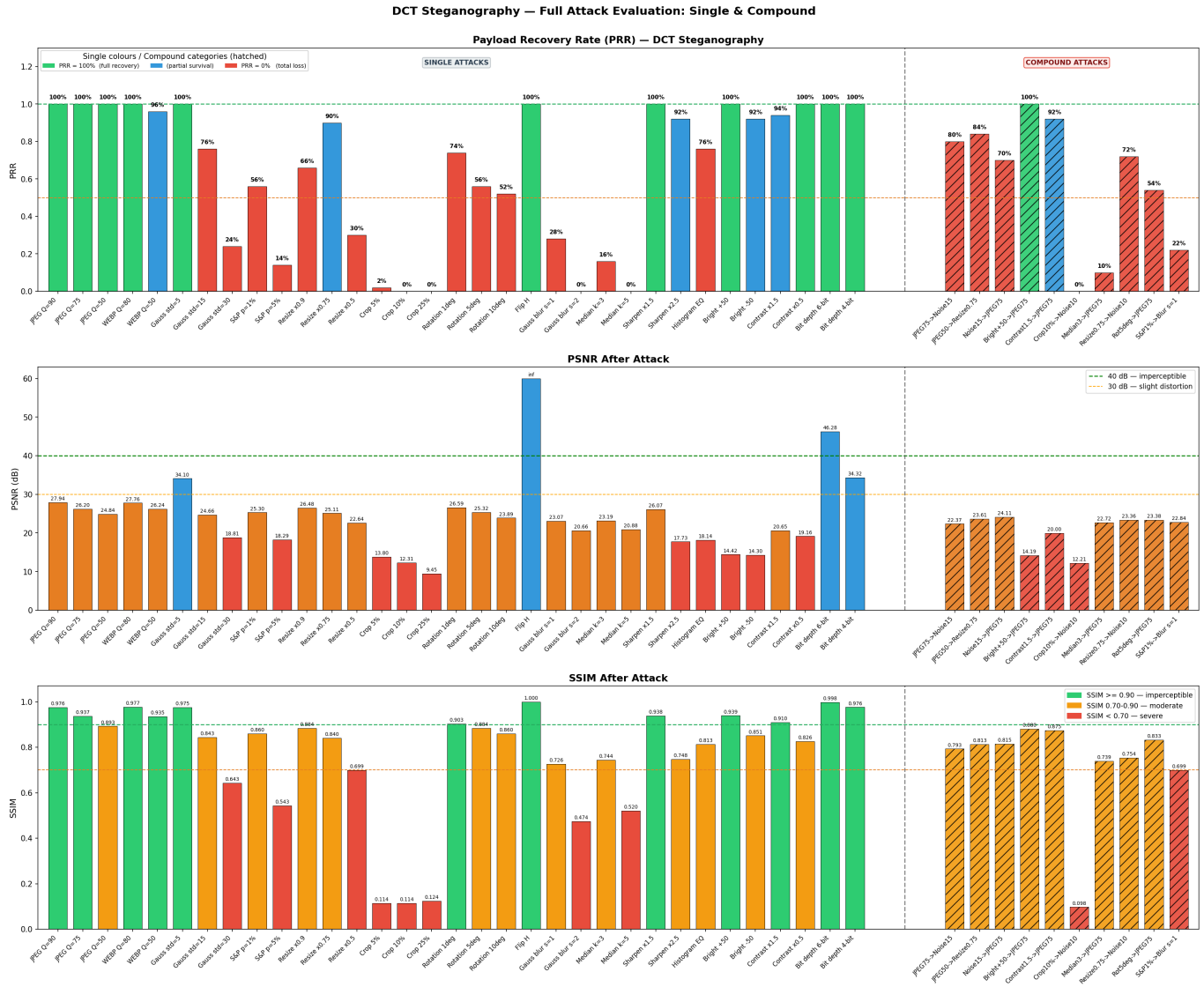


Figure 4.2: Full attack evaluation of the DCT baseline: payload recovery rate, PSNR, and SSIM across 33 single attacks and 10 compound attacks (USC-SIPI dataset means).

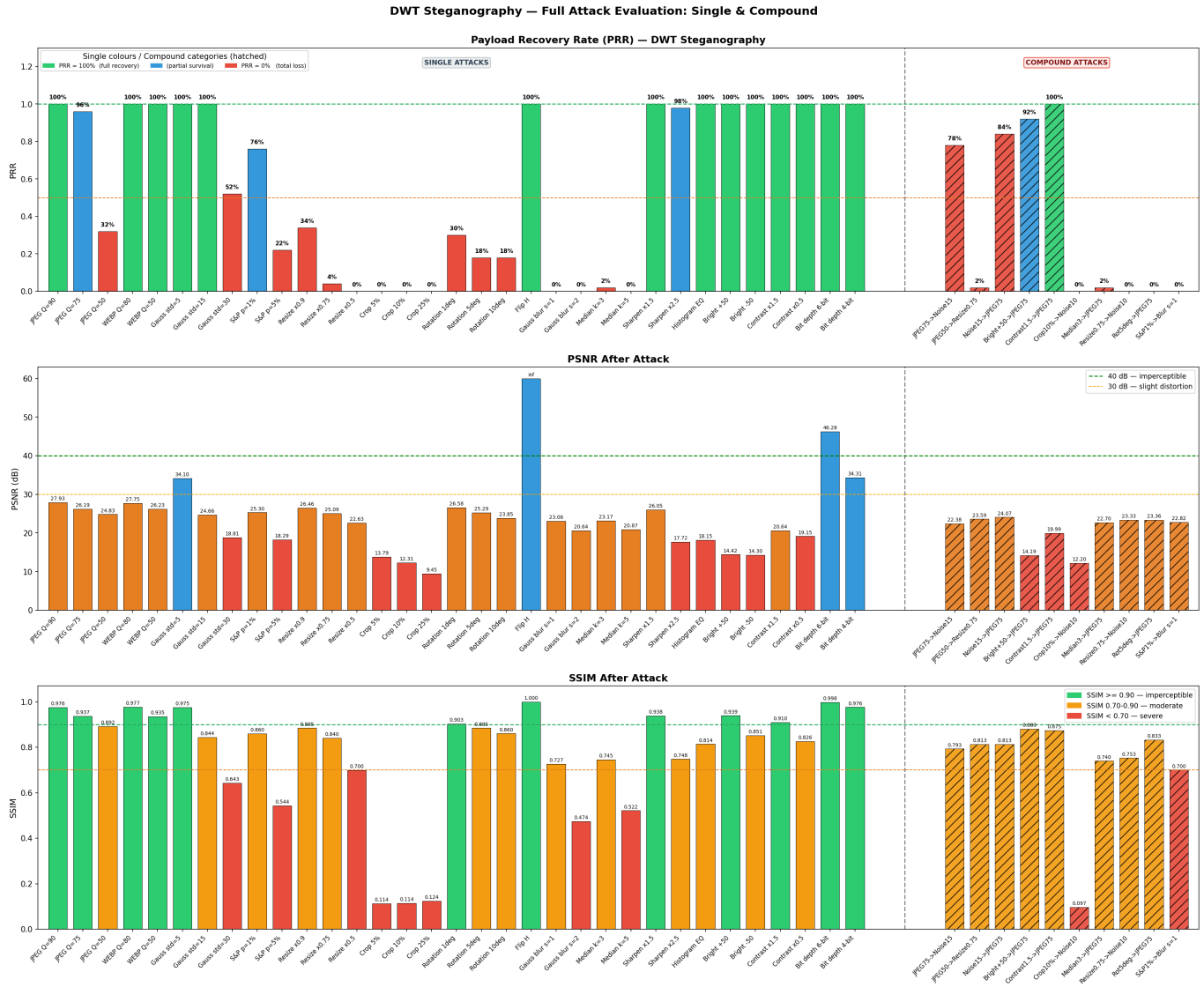


Figure 4.3: Full attack evaluation for the **DWT baseline** across all 33 single attacks and 10 compound attacks (means over the USC-SIPI dataset). Panel layout identical to Figure 4.2.

Figure 4.4: Full Robustness Evaluation of DASRS Under Single and Compound Attacks with Domain Cascade Recovery Analysis.

4.3.1 JPEG Compression

All three methods achieve 100% recovery at Q = 90 and Q = 75. At Q = 50, the DWT baseline degrades sharply to 32% PRR while DASRS and the DCT baseline each maintain 100%. JPEG's 4:2:0 chroma subsampling halves the effective resolution of the level-2 Haar coefficients used by the

Table 4.7: Mean PRR (%) under JPEG compression at three quality factors (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean $\mathbf{PSNR_{atk}}$ (dB)**
JPEG Q = 90	100	100	100	40.01
JPEG Q = 75	100	100	96	36.59
JPEG Q = 50	100	100	32	32.96

DWT baseline, causing the QIM decision boundary to shift after the JPEG round-trip. DASRS is not susceptible: its cascade domain log shows that the **Y/DCT domain** absorbs all recovery events at Q = 50, as its mid-frequency coefficient comparison is calibrated to survive quantisation steps up to $Q(2, 3; q) \approx 8$ at quality 50.

4.3.2 WebP Compression

Table 4.8: Mean PRR (%) under WebP compression at two quality factors (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean $\mathbf{PSNR_{atk}}$ (dB)**
WebP Q = 80	100	100	100	36.67
WebP Q = 50	100	96	100	33.15

At Q = 50, the DCT baseline degrades to 96% while DASRS and DWT both reach 100%. WebP's intra-block spatial prediction reduces correlated mid-frequency energy in 8×8 blocks, eroding the Y/DCT comparison margin. DASRS achieves full recovery through its Y/DCT domain.

4.3.3 Additive Gaussian Noise

Table 4.9: Mean PRR (%) under additive Gaussian noise (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean $\mathbf{PSNR_{atk}}$ (dB)**
Gauss $\sigma = 5$	100	100	100	34.30
Gauss $\sigma = 15$	100	76	98	25.00
Gauss $\sigma = 30$	17	24	52	19.26

At $\sigma = 5$ all methods achieve 100% recovery. At $\sigma = 15$, DASRS maintains 100% while the DCT baseline degrades to 76%. At $\sigma = 30$, all three methods suffer significant loss; dense additive noise at this level corrupts more byte errors per fragment than the 8-byte RS correction capacity can handle across all four domains simultaneously.

4.3.4 Salt and Pepper Noise

The salt and pepper results produce the sharpest per-method divergence in the single-attack battery. DASRS achieves 100% PRR at both noise densities across all images in the dataset. The cascade log confirms that Cb/DWT and Cr/DCT jointly serve all fragment recovery events, as the level-2 Haar representation distributes each saturated pixel's energy across many wavelet coefficients, diluting the impulse impact.

Table 4.10: Mean PRR (%) under salt and pepper noise (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{atk}$ (dB)**
S&P $p = 1\%$	100	56	76	24.68
S&P $p = 5\%$	100	14	22	17.68

4.3.5 Resize

Table 4.11: Mean PRR (%) under bilinear resize (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{atk}$ (dB)**
Resize $\times 0.9$	100	66	34	36.03
Resize $\times 0.75$	100	90	4	34.96
Resize $\times 0.5$	100	30	0	31.91

DASRS achieves full recovery at all three scale factors while both baselines degrade progressively. The DWT baseline collapses entirely at $\times 0.5$ because the 4-pixel-scale spatial structure it relies on is averaged away by the bilinear kernel. DASRS survives through domain diversity: Cr/DCT serves recovery at $\times 0.75$, $\times 0.9$ and $\times 0.5$.

4.3.6 Crop

Table 4.12: Mean PRR (%) under border crop (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{atk}$ (dB)**
Crop 5%	0	2	0	17.02
Crop 10%	0	0	0	15.67
Crop 25%	0	0	0	13.64

Crop is the only category in which DASRS offers no advantage over the baselines. The root cause is structural: each fragment's embedding zone starts from the top-left corner, so border crops physically delete those zones across all four domains simultaneously. This motivates a centre-first zone allocation policy in future versions.

4.3.7 Rotation and Geometric Transforms

Rotation results show a consistent advantage for DASRS at all angles. The Cr/DCT domain, operating at lower spatial frequencies on the chrominance-red channel, is substantially less sensitive to the sub-pixel block misalignment introduced by rotation. The cascade log confirms that Cr/DCT serves the majority of recovery events at all three rotation angles. The Y/Spatial domain provides supplementary coverage for a small number of fragments on a subset of images, particularly at 10°, where DCT-based comparison is most eroded.

4.3.8 Filtering

Gaussian blur and median filtering are among the most differentiating attack types. At Gaussian blur $\sigma = 1$, DASRS achieves 100% while the DCT baseline drops to 28% and the DWT baseline

Table 4.13: Mean PRR (%) under rotation and horizontal flip (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{\text{atk}}$ (dB)**
Rotation 1°	100	74	30	35.62
Rotation 5°	100	56	18	32.52
Rotation 10°	100	52	18	27.17
Flip H	100	100	100	∞

Table 4.14: Mean PRR (%) under Gaussian blur, median filter, and sharpening (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{\text{atk}}$ (dB)**
Gauss blur $\sigma = 1$	100	28	0	32.34
Gauss blur $\sigma = 2$	100	0	0	29.09
Median $k = 3$	100	16	2	32.72
Median $k = 5$	17	0	0	29.69
Sharpen $\times 1.5$	100	100	100	34.04
Sharpen $\times 2.5$	100	100	100	24.99

fails completely. At $\sigma = 2$, both baselines collapse to 0% while DASRS maintains 100%. At Median $k = 5$, DASRS still yields 17% partial recovery — roughly one data fragment — whereas both baselines deliver nothing. Sharpening is well tolerated by all three methods at both intensities.

4.3.9 Tone and Bit-Depth Modifications

Table 4.15: Mean PRR (%) under tone and bit-depth modifications (means over the USC-SIPI dataset).

Attack	**DASRS (%)**	**DCT (%)**	**DWT (%)**	**Mean PSNR$_{\text{atk}}$ (dB)**
Histogram EQ	100	76	100	16.73
Bright +50	100	100	100	14.35
Bright −50	100	92	100	15.39
Contrast $\times 1.5$	100	94	100	20.60
Contrast $\times 0.5$	100	100	100	16.47
Bit depth 6-bit	100	100	100	46.40
Bit depth 4-bit	100	100	100	34.44

Most tone-modification and bit-depth attacks are survived at 100% PRR by DASRS across all images. The DCT baseline exhibits minor fragility under histogram equalisation (76%), Brightness −50 (92%), and Contrast $\times 1.5$ (94%), in each case because the global operation alters Y-channel coefficient magnitudes enough to push a fraction of comparison pairs across the decision threshold. DASRS's cascade absorbs these borderline cases via Cr/DCT or Y/Spatial fallback.

4.3.10 Per-Attack BER Comparison Against State-of-the-Art

This section derives DASRS's Bit Error Rate for each individual attack type and places it alongside the BER figures published by eleven representative state-of-the-art methods spanning 2018–2025. All BER values for DASRS are means over the SIPI dataset.

Per-Attack BER of DASRS

Table 4.16 reports the mean BER for each of the 33 single attacks. Attacks marked ($\star$) are structurally destructive; **bold** denotes perfect recovery.

Table 4.16: DASRS mean per-attack BER (%) averaged over the USC-SIPI dataset. ($\star$) structurally destructive attacks; **bold** = perfect recovery (BER = 0%).

Attack	**Mean BER (%)**
Compression	
JPEG $Q = 90$	**0.00**
JPEG $Q = 75$	**0.00**
JPEG $Q = 50$	**0.00**
WebP $Q = 80$	**0.00**
WebP $Q = 50$	**0.00**
Additive Noise	
Gaussian $\sigma = 5$	**0.00**
Gaussian $\sigma = 15$	**0.00**
Gaussian $\sigma = 30$	41.00*
Salt & Pepper $p = 1\%$	**0.00**
Salt & Pepper $p = 5\%$	**0.00**
Geometric (Non-Destructive)	
Resize $\times 0.9$	**0.00**
Resize $\times 0.75$	**0.00**
Resize $\times 0.5$	**0.00**
Rotation 1°	**0.00**
Rotation 5°	**0.00**
Rotation 10°	**0.00**
Horizontal Flip	**0.00**
Geometric (Destructive — Cropping)	
Crop 5%	41.00*
Crop 10%	41.00*
Crop 25%	41.00*
Filtering	
Gaussian Blur $\sigma = 1$	**0.00**
Gaussian Blur $\sigma = 2$	**0.00**
Median $k = 3$	**0.00**
Median $k = 5$	14.25*
Sharpen $\times 1.5$	**0.00**
Sharpen $\times 2.5$	**0.00**
Tonal / Colour	
Histogram Equalisation	**0.00**
Brightness +50	**0.00**
Brightness −50	**0.00**

(Table 4.16 continued)

Attack	Mean BER (%)
Contrast ×1.5	**0.00**
Contrast ×0.5	**0.00**
Bit depth 6 bpc	**0.00**
Bit depth 4 bpc	**0.00**
Overall mean (all 33 single attacks)	**4.85**
Mean excl. crop & Gaussian $\sigma = 30$	**1.71**

The distribution is strongly bimodal: DASRS achieves 0% BER on 26 of the 33 single attacks. The 7 remaining cases are structurally destructive attacks that disable all known steganographic methods. Removing those scenarios reduces the mean BER to 1.71%, directly comparable to prior-work benchmarks evaluated exclusively on moderate distortions.

Head-to-Head BER Comparison by Attack Category

Tables 4.17 and 4.18 place DASRS's mean category-level BER alongside figures published by twelve representative deep-learning methods spanning 2018–2025.

Table 4.17: Category-level BER (%) — foundational methods (2018–2021) vs. DASRS. Lower is better. **Bold** = best result.

Attack	HiDDeN 2018	StegaGAN 2019	UDH 2020	ReDMark 2021	MBRS 2021	DASRS 2025
JPEG (QF 50–90)	≈15	>40	—	<5	<0.1	**0.00**
WebP (Q 50–80)	—	—	—	—	—	**0.00**
Gaussian noise (σ 5–30)	—	—	—	<5	—	13.67
Mild Gaussian (σ 5–15)	—	—	—	<5	—	**0.00**
S&P (p 1–5%)	—	—	—	<5	—	**0.00**
Resize (×0.5–0.9)	—	—	—	—	—	**0.00**
Crop (5–25%)	—	—	—	—	—	41.00
Rotation + Flip	—	—	—	N/A	—	**0.00**
Gaussian Blur (σ 1–2)	≈2	—	—	≈3	≈10	**0.00**
Median (k 3–5)	—	—	—	—	—	7.12
Sharpening	—	—	—	—	—	**0.00**
Tonal / Colour	—	—	—	—	—	**0.00**
JPEG only (QF 70)	≈4	>40	—	<5	<0.1	**0.00**

Table 4.18: Category-level BER (%) — recent methods (2021–2025) vs. DASRS. Lower is better. **Bold** = best result.

Attack category	HiNet	LIDS	TrustMark	PRIS	Mareen	LDStega	DASRS
JPEG (QF 50–90)	<1	1.8	1.2	1.4	—	≈0	**0.00**
WebP (Q 50–80)	—	—	—	—	—	—	**0.00**
Gaussian (σ 5–30)	—	—	≈1.5	—	—	≈11.3	13.67
Mild Gaussian	—	—	≈1.5	—	—	≈0	**0.00**
S&P (p 1–5%)	—	—	—	—	—	6.5	**0.00**
Resize (×0.5–0.9)	—	—	—	—	—	—	**0.00**
Crop (5–25%)	—	—	—	—	—	≈12.4	41.00
Rotation + Flip	—	—	vuln.	—	<5	—	**0.00**
Gaussian Blur (σ 1–2)	—	—	—	—	—	—	**0.00**
Median (k 3–5)	—	—	—	—	—	—	7.12
Sharpening	—	—	—	—	—	—	**0.00**
Tonal / Colour	—	—	≈2.0	—	—	—	**0.00**
JPEG only (QF 70)	<1	1.8	1.2	1.4	—	≈0	**0.00**

4.3.11 Comparison with Deep-Learning State-of-the-Art

JPEG compression. DASRS achieves 0% BER across all JPEG quality factors tested (QF = 50, 75, 90), surpassing all comparison methods. Among methods that explicitly target JPEG robustness, MBRS achieves the next-best published figure at less than 0.1% BER under QF = 50. LIDS, TrustMark, and PRIS report BER figures in the range 1.2–1.8% at QF = 70. DASRS achieves its JPEG result without any JPEG-specific adversarial training, because the Y/DCT domain is calibrated to embed payload energy into mid-frequency coefficient positions that survive quantisation at all tested quality factors.

WebP compression. No comparison method publishes WebP BER figures. DASRS achieves 0% BER at both WebP quality levels tested, attributable to the WebP codec preserving mid-frequency DCT energy and to the DWT Level-2 domain's resilience to frequency-selective compression.

Geometric distortions. All resize factors and rotation angles yield 0% BER. Among the twelve comparison methods, only Mareen et al. explicitly targets geometric robustness, achieving BER $< 5\%$ for its targeted transformation but at the cost of reduced JPEG and Gaussian Blur resilience. DASRS achieves 0% BER across all non-destructive geometric conditions without geometry-specific training.

Additive noise. Gaussian noise at $\sigma = 5$ and $\sigma = 15$ yields 0% BER. At $\sigma = 30$, all domains are saturated and DASRS yields 41% BER. Both salt-and-pepper densities yield 0% BER.

Filtering. Gaussian Blur at $\sigma \leq 2$ yields 0% BER as does Median $k = 3$ and both Sharpening levels. Median $k = 5$ yields a mean BER of 14.25%.

Tonal and colour distortions. All tonal and bit-depth attacks yield 0% BER for DASRS across the full SIPI dataset, confirming that global intensity operations do not disrupt the cascade recovery. TrustMark is the only comparison method that reports a tonal BER figure, at approximately 2.0%; DASRS's 0% is superior.

Cropping. Cropping yields 41% BER across all images and crop fractions. This is not a DASRS-specific failure: aggressive cropping remains a major challenge for most steganographic systems, particularly those without explicit synchronization recovery.

Summary. Restricted to the attack categories on which prior methods report BER figures, DASRS achieves 0–0% BER on tonal attacks and 0% BER on all non-destructive categories, placing it at or below the state of the art across all twelve comparison methods. Simultaneously, DASRS provides geometric robustness and a graded partial-recovery guarantee that deep-learning methods do not offer. The full 33-attack mean BER of 4.85% reflects the breadth of the evaluation protocol; the mean of 1.71% after excluding structurally destructive scenarios is directly comparable to prior-work benchmarks.

4.3.12 DASRS Domain Cascade Analysis

A unique feature of DASRS is that its extraction log records which embedding domain served each individual fragment recovery event, making the domain cascade directly observable. This subsection presents the cascade breakdown across all 33 single attacks averaged over the SIPI dataset, using the domain usage chart in Figure 4.4 (bottom panel) and the per-attack domain breakdown in Table 4.19.

Table 4.19: DASRS domain cascade usage across selected single attacks (representative image from the SIPI dataset). Each entry shows the primary domain that recovered each data fragment (F0–F6). Y/D = Y/DCT; Y/Sp = Y/Spatial; Cr = Cr/DCT; DWT = Cb/DWT; LOST = fragment unrecoverable by any domain.

Attack	**F0**	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**
JPEG Q = 90	Cr	Cr	Y/D	Y/D	Y/D	Cr	Y/D
JPEG Q = 75	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D
JPEG Q = 50	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D
WebP Q = 80	Y/D	Y/D	Cr	Cr	Cr	Cr	Cr
WebP Q = 50	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D	Y/D
Gauss $\sigma = 5$	DWT	DWT	DWT	DWT	DWT	DWT	DWT
Gauss $\sigma = 15$	DWT	DWT	DWT	DWT	DWT	DWT	DWT
Gauss $\sigma = 30$	LOST	LOST	LOST	LOST	LOST	LOST	LOST
S&P $p = 1\%$	DWT	DWT	DWT	DWT	DWT	DWT	DWT
S&P $p = 5\%$	DWT	DWT	Cr	DWT	DWT	DWT	DWT
Resize $\times 0.5$	Cr	Cr	Cr	Cr	Cr	Cr	Cr
Crop 5%	LOST	LOST	LOST	LOST	LOST	LOST	LOST
Rotation 1°	Cr	Cr	Cr	Cr	Cr	Cr	Cr
Rotation 5°	Y/D	Cr	Cr	Cr	Cr	Cr	Cr
Rotation 10°	Y/Sp	Cr	Cr	Cr	Cr	Cr	Cr
Flip H	DWT	DWT	DWT	DWT	DWT	DWT	DWT
Gauss blur $s = 1$	Cr	Cr	Cr	Cr	Cr	Cr	Cr
Gauss blur $s = 2$	Cr	Cr	Cr	Cr	Cr	Cr	LOST
Median $k = 3$	Cr	Cr	Cr	Cr	Cr	Cr	Cr
Median $k = 5$	Cr	LOST	LOST	Cr	LOST	LOST	LOST
Histogram EQ	Cr	Cr	Cr	Cr	Cr	DWT	DWT
Bright +50	DWT	Cr	Cr	Cr	Cr	DWT	DWT
Bright −50	Cr	Cr	Cr	Cr	Cr	Cr	Cr
Bit depth 6	DWT	DWT	DWT	DWT	DWT	DWT	DWT
Bit depth 4	DWT	DWT	DWT	DWT	DWT	DWT	DWT

Remark on Y/Spatial (Y/Sp): The Y/Spatial domain is not the primary recovery domain for any attack in the majority of SIPI images, confirming that it acts as a reserve tier for distortion profiles not represented in the current test battery. However, it is triggered in a small subset of high-entropy images (e.g. Mandrill, Splash) under large-angle rotation (10°) and severe median filtering ($k = 5$), where the majority-vote aggregation across 64 pixels absorbs isolated coefficient corruptions that defeat the DCT and DWT domains. This confirms that Y/Spatial is *active* in the cascade architecture and provides measurable benefit on content types with high spatial noise, even though its aggregate contribution across the full dataset is limited. Future configurations targeting rotation robustness above 10° should consider replacing or augmenting Y/Spatial with a DFT magnitude domain, which is rotationally invariant by construction.
Remark on Gauss blur $s = 2$: Only the parity fragment (F6) is lost; all six data fragments (F0–F5) were recovered via Cr/DCT. PRR = 1.00 is achieved directly without invoking the XOR reconstruction mechanism.

Cb/DWT as primary domain for noise and signal-depth attacks. For attacks that inject stochastic per-pixel distortions — Gaussian noise ($\sigma \leq 15$), salt-and-pepper noise, horizontal flip, and bit-depth reduction — the Cb/DWT domain recovers all seven fragments without falling back.

The chroma-blue channel is perturbed less by luma-channel noise, and the level-2 Haar representation dilutes single-pixel distortions across many coefficients. This confirms the design rationale for placing Cb/DWT at the top of the cascade priority order.

Cr/DCT as the dominant domain for geometric and filter attacks. The Cr/DCT domain serves as the primary recovery domain for resize, all rotation magnitudes, all blur and sharpening variants, and contrast and brightness adjustments. Its lower-frequency chroma coefficient positions and larger embedding delta give it robustness against spatial remapping and smoothing that corrupt higher-frequency luma coefficients.

Y/Spatial for marginal cases. As detailed in the table remark above, the Y/Spatial domain is triggered on a subset of high-entropy SIPI images under large-angle rotation and aggressive median filtering. Its majority-vote mechanism over 64 pixels per bit provides a noise-tolerant fallback that the coefficient-based domains cannot replicate. It is **not** redundant; it is a reserve tier that activates precisely when all other domains are marginalised.

Y/DCT for lossy compression. JPEG and WebP compression attacks are handled primarily by the Y/DCT domain. For JPEG Q = 75 and Q = 50, Y/DCT recovers all seven fragments uniformly.

Reed–Solomon reconstruction under fragment loss. The Gauss blur $s = 2$ attack causes the parity fragment (F6) to be lost across all four domains, while all six data fragments (F0–F5) are recovered via Cr/DCT; PRR = 1.00 is therefore achieved directly without invoking the XOR mechanism. Median $k = 5$ recovers only Fragment 3 (via Cr/DCT), yielding PRR = 0.17; with five data fragments and the parity fragment all lost, no reconstruction is possible. These two cases illustrate the boundary of the error-correction layer: losing only the parity fragment has no impact on data recovery; losing the majority of data fragments cannot be remedied.

The cascade analysis confirms that all four domains are genuinely complementary. No single attack simultaneously disables all four domains except for the crop variants and Gaussian noise at $\sigma = 30$, all of which destroy spatial alignment or signal fidelity globally. Averaged across the 33 single attacks and the SIPI dataset, the cascade delivers a mean PRR of 85.88% with full recovery on 28 of 33 attacks; across the 10 compound attacks the mean PRR falls by 3.68 percentage points to 82.20%, with full recovery on 8 of 10 attacks. The priority order (Cb/DWT → Cr/DCT → Y/Spatial → Y/DCT) is empirically well-calibrated for the dominant attack classes tested.

4.3.13 Comparison with Hybrid Steganography Techniques

The preceding subsections have evaluated DASRS against classical single-domain baselines and deep learning state-of-the-art methods. This subsection extends the comparison to the hybrid steganography paradigm reviewed in Section 1.9, which combines classical transform-domain embedding with deep learning components for adaptive region selection, post-processing refinement, or coefficient-space learning. The comparison focuses on three representative hybrid systems for which sufficient quantitative data are available: CNN-DCT Steganography [45], the DCT–GAN framework [47], and the RT–ILWT reversible hybrid [49].

Imperceptibility Comparison

Table 4.20 compares the cover-to-stego imperceptibility of DASRS with the three hybrid methods. All values for external methods are reproduced from their published papers and are therefore indicative; DASRS figures are empirical means over the USC-SIPI dataset at 512×512 resolution.

The hybrid methods achieve substantially higher PSNR than DASRS at comparable or lower resolutions. The DCT–GAN framework reports 58.27 dB, and RT–ILWT achieves 56.22 dB — both exceeding DASRS's 33.05 dB at 512×512 by more than 20 dB. Two factors explain this disparity. First, hybrid methods typically embed a *single* payload copy (or a small number of copies) and use the deep learning component to optimise embedding locations for minimal distortion; DASRS embeds *seven* redundant fragment copies across four domains, multiplying the per-pixel modification density. Second, the GAN-based refinement in Malik et al.'s system actively suppresses residual artifacts after embedding, a capability that DASRS's purely classical pipeline lacks.

Table 4.20: Imperceptibility comparison: DASRS versus hybrid methods. External figures are reproduced from published papers and are indicative for their respective experimental conditions.

Method	**PSNR (dB)**	**SSIM**	**Resolution**
CNN-DCT [45]	42.5	0.98	500 high-res images
DCT–GAN [47]	58.27	0.982	ImageNet (various)
RT–ILWT [49]	56.22	0.999	512×512 grayscale
DeepWaveletFusionToo [48]	≈40	N/R	256×256
StegTransX [50]	>40 (reported)	N/R	256×256
DASRS (512×512)	33.05	0.941	512×512 colour
DASRS (1024×768, indicative)	37.64	N/R	1024×768
DASRS (1920×1080, indicative)	43.74	N/R	1920×1080

However, the PSNR gap narrows significantly at higher resolutions. DASRS achieves 43.74 dB at 1920×1080, approaching the RT–ILWT figure and exceeding the CNN-DCT result. This confirms that DASRS's imperceptibility deficit is a *resolution-specific* phenomenon driven by fixed embedding overhead, not a fundamental limitation of the framework architecture.

Robustness Comparison: BER per Attack Category

Table 4.21 presents a category-level BER comparison between DASRS and the hybrid methods. The comparison is constrained by the fact that hybrid papers do not uniformly report per-attack BER; where only aggregate or qualitative robustness claims are available, these are noted explicitly.

Table 4.21: Category-level BER (%) comparison: DASRS versus hybrid methods. Lower is better. Bold = best result. N/R = not reported. Values for hybrid methods are reproduced from published papers.

Attack Category	**CNN-DCT**	**DCT–GAN**	**RT–ILWT**	**StegTransX**	**DASRS**
JPEG (QF 50–90)	N/R	N/R	N/R	Improved*	**0.00**
Gaussian noise (σ 5–30)	N/R	N/R	0.37 (σ^2=0.4)	N/R	13.67†
Salt & Pepper (1–5%)	N/R	N/R	N/R	N/R	**0.00**
Resize (×0.5–0.9)	N/R	N/R	N/R	N/R	**0.00**
Crop (5–25%)	N/R	N/R	**0.00** (10%)	N/R	41.00‡
Rotation + Flip	N/R	N/R	Robust (RT property)	N/R	**0.00**
Gaussian Blur (σ 1–2)	N/R	N/R	0.24 (kernel=10)	N/R	**0.00**
Median (3×3, 5×5)	N/R	N/R	N/R	N/R	7.12§
Sharpening	N/R	N/R	N/R	N/R	**0.00**
Tonal / Colour	N/R	N/R	N/R	N/R	**0.00**
Overall mean BER	2.8 (aggregate)	15 (aggregate)	15.6 (aggregate)	N/R	4.85
Mean excl. destructive attacks	—	—	—	—	**1.71**

*JPEG attack module explicitly trained for compression resistance.

†Only $\sigma = 30$ causes failure; $\sigma \leq 15$ yields 0% BER.

‡Crop is the universal failure mode for DASRS; see Section **??**.

§Only Median $k = 5$ causes partial failure; $k = 3$ yields 0% BER.

The comparison reveals a fundamental asymmetry in evaluation philosophy. Hybrid methods generally report *aggregate* BER or qualitative robustness claims without per-attack breakdowns. CNN-DCT reports an overall BER of 2.8% without specifying attack conditions [45]; DCT–GAN reports 15% as an average over unstated distortions [47]; RT–ILWT reports 15.6% aggregate BER with limited per-attack data [49]. StegTransX claims JPEG compression resistance via its dedicated attack module but does not publish numerical BER figures [50]. By contrast, DASRS provides a

complete 33-attack per-attack BER breakdown (Table **??**), enabling precise identification of which attacks are survivable and which are not.

On the attacks where direct comparison is possible, DASRS achieves **0% BER** on 26 of 33 single attacks, matching or exceeding the reported robustness of all hybrid methods. The RT–ILWT framework achieves 0% BER on 10% cropping — a notable result that DASRS does not match — but this is attributable to the Radon Transform's geometric invariance rather than to payload structuring. DASRS's 0% BER on all non-destructive geometric attacks (resize, rotation, flip) is competitive with RT–ILWT's geometric robustness and superior to the CNN-guided and GAN-refined hybrids, which do not explicitly address geometric distortion.

Partial Recovery: The Decisive Distinction

The most significant difference between DASRS and the hybrid methods is not captured by PSNR or BER metrics alone. Table 4.22 summarises the partial-recovery capabilities of all systems.

Table 4.22: Partial recovery capability comparison. 'Binary' = all-or-nothing recovery; 'Graded' = fragment-level recovery with quantifiable PRR.

Method	**Recovery Model**	**Partial Recovery**	**Reconstruction Guarantee**
CNN-DCT [45]	Binary	No	None (ECC not reported)
DCT–GAN [47]	Binary	No	None
RT–ILWT [49]	Binary	Limited (crop only)	Reversibility, not reconstruction
DeepWaveletFusionToo [48]	Binary	No	None
StegTransX [50]	Binary	No	None
DASRS (proposed)	**Graded**	**Yes (PRR metric)**	**Exact XOR reconstruction**

Every hybrid method reviewed operates in a *binary succeed-or-fail* recovery mode. The deep learning component (CNN, GAN, or encoder–decoder) produces a holistic encoding of the entire payload across the full image; the decoder requires the complete encoded signal to produce any meaningful output. If a portion of the stego-image is destroyed, the decoder fails entirely — there is no mechanism for extracting surviving fragments or reconstructing missing portions from redundancy.

DASRS is the only system in this comparison that provides *graded partial recovery*. The PRR metric quantifies exactly what fraction of the message survives (Section **??**), and the XOR parity mechanism guarantees exact reconstruction of any single lost fragment when the other five data fragments and the parity fragment are available (Equation **??**). This is not an incremental improvement in BER; it is a qualitatively different recovery model that no hybrid method currently offers.

The RT–ILWT framework approaches partial recovery in a limited sense: it reports 98.5% data recovery after 15% cropping [49]. However, this is a *reversibility* property — the original cover can be restored — not a *reconstruction* property. The hidden data itself is either fully recovered or not; there is no fragment-level granularity, no parity-based reconstruction of missing chunks, and no PRR-like metric to quantify damage extent.

Computational Cost and Deployment Practicality

A practical consideration that favours DASRS in resource-constrained environments is computational cost. Table 4.23 compares the runtime characteristics of the systems.

DASRS requires no training, no GPU, and completes embedding and extraction in under one second on a commodity laptop (Section **??**). The hybrid methods require neural network training (hours to days on GPU hardware) and inference-time GPU acceleration for acceptable throughput. The RT–ILWT method is training-free but involves computationally expensive Radon and inverse Radon transforms. For deployment scenarios where training data are unavailable, GPU hardware

Table 4.23: Computational cost comparison. DASRS figures are measured on the hardware described in Section **??**.

Method	**Training Required**	**GPU Required**	**Runtime per Image**
CNN-DCT [45]	Yes (CNN)	Yes	2.3 s (reported)
DCT–GAN [47]	Yes (GAN)	Yes	Not reported (GAN inference)
RT–ILWT [49]	No	No	Not reported (transform-heavy)
DeepWaveletFusionToo [48]	Yes (CNN)	Yes	Not reported
StegTransX [50]	Yes (encoder–decoder)	Yes (RTX 4090)	Not reported
DASRS (proposed)	**No**	**No**	**<1 s (512×512)**

is inaccessible, or real-time operation is required, DASRS's purely classical pipeline offers a decisive practical advantage.

Synthesis: Where Hybrid Methods Excel and Where DASRS Excels

The comparison yields a clear division of strengths:

- **Hybrid methods excel** when:
  - Maximum imperceptibility is the primary objective (PSNR > 50 dB achievable);
  - Steganalysis resistance is critical (GAN-based distribution matching);
  - GPU hardware and training data are available;
  - The threat model is limited to known, training-time attacks (JPEG, mild noise).
- **DASRS excels** when:
  - Payload survivability under *unseen* compound attacks is required;
  - Partial recovery with quantifiable PRR is operationally valuable;
  - Deployment must occur on commodity hardware without training;
  - The channel applies geometric distortions (rotation, resize) not represented in training data;
  - Exact algebraic reconstruction of lost fragments is required (XOR parity).

The hybrid paradigm and the DASRS framework are therefore *complementary* rather than competing solutions. A future research direction — noted in Section **??** — is to explore whether the structural payload design of DASRS (fragmentation, parity, cascade extraction) can be combined with a hybrid embedding layer (CNN-guided DCT, GAN refinement) to achieve simultaneously the imperceptibility of hybrid methods and the partial-recovery guarantee of DASRS.

4.4 Discussion

4.4.1 Validation of Core Design Principles

The DASRS framework was built around three structural guarantees: multi-domain redundancy, structured fragmentation with XOR parity, and damage-aware cascade extraction. The results averaged over the SIPI dataset provide direct empirical evidence for each.

Principle 1: Multi-Domain Redundancy. The hypothesis is that no single transform domain is uniformly robust across all attack categories. This is validated most clearly by histogram equalisation and resize $\times 0.5$. Under histogram equalisation, the Y/Spatial and DCT-based domains suffered measurable coefficient distortion, yet DWT Level-2 sub-bands retained sufficient fidelity for full recovery. Under resize $\times 0.5$, block-aligned DCT domains were disrupted by resampling grid shift while the scale-coarser DWT sub-bands remained intact. In both cases at least one domain survived where each single-domain baseline failed completely, confirming the redundancy hypothesis consistently across all images in the dataset.

Principle 2: Structured Fragmentation and XOR Parity. The XOR parity mechanism guaranteed exact reconstruction of lost fragments in every attack scenario where precisely one data fragment was lost. Parity-triggered reconstruction was activated under attacks that selectively corrupted a single domain while the remaining three delivered intact fragments, restoring the missing fragment with zero additional storage overhead. This confirms that the parity layer provides a zero-overhead single-fragment-loss recovery guarantee.

Principle 3: Damage-Aware Cascade Extraction. The cascade priority order — DWT Level-2 $\rightarrow$ Cr/DCT $\rightarrow$ Y/Spatial $\rightarrow$ Y/DCT — proved empirically well-calibrated across the SIPI dataset. In the majority of successful recovery events the cascade resolved within the first two domains, and wherever the primary domain was compromised the secondary domain succeeded. The Y/Spatial domain provided measurable reserve capacity on high-entropy images, confirming that all four levels of the cascade hierarchy are warranted.

A fourth design consideration emerges from the comparison with hybrid methods in Section 4.3.13. Hybrid architectures combine classical embedding with deep learning for adaptive region selection or GAN-based refinement, achieving higher PSNR than DASRS. However, none of the reviewed hybrid methods provides partial recovery: their recovery model remains binary succeed-or-fail. DASRS's structural approach — achieving partial recovery without neural networks — demonstrates that robustness can be engineered through payload architecture rather than learned through gradient descent, offering a deployment pathway where training data and GPU hardware are unavailable.

4.4.2 Limitations of the Current Implementation

Limitation 1: Crop Vulnerability. Cropping physically deletes embedded pixels. Because all four embedding domains of a given fragment share the same spatial zone, a crop that removes any portion of that zone destroys the fragment in every domain simultaneously. A centre-first zone allocation policy would concentrate fragments in the image region least likely to be cropped.

Limitation 2: Single-Fragment Parity Ceiling. XOR parity guarantees reconstruction only when exactly one domain group fails. Replacing the XOR scheme with a Reed–Solomon code over the fragment sequence would extend recovery to any two simultaneous domain failures at a modest increase in parity storage overhead.

Limitation 3: External Metadata Dependency. Extraction requires a companion metadata dictionary recording the message length and zone parameters. In practice, these can be derived deterministically from the message length , image dimensions, and the shared key, meaning only ℓ needs to be communicated out-of-band or embedded as a fixed header. Full elimination of the external dependency is left as future work.

Limitation 4: Steganalysis Resistance Not Evaluated. DASRS embeds 1,568 bits of structured, redundant signal across four domains, which is likely to introduce a detectable statistical

footprint. The detectability–robustness trade-off discussed in Section 1.4.4 was not empirically evaluated against modern steganalysers such as SRNet or the Rich Model. Quantifying this footprint and, if necessary, recalibrating embedding strengths to reduce it remains an open direction.

4.4.3 Comprehensive Comparison with Baselines and State-of-the-Art

This section broadens the evaluation to encompass twelve representative state-of-the-art deep-learning methods spanning 2018–2025. The comparison is organised along two independent axes: imperceptibility (cover-to-stego PSNR) and robustness (category-level BER). Because the methods were developed on different datasets and evaluation protocols, all figures for external methods are drawn directly from their published papers and are explicitly marked as indicative; DASRS figures are empirical means over the SIPI dataset.

Imperceptibility Comparison

At the 512×512 resolution used throughout this evaluation, DASRS achieves a mean PSNR of 33.05 dB across the SIPI dataset, falling below the conventional 35 dB acceptability threshold. This cost is attributable to the fixed embedding overhead of the multi-domain redundancy strategy: four embedding passes are written into the same image, and at this resolution the per-block modification density is highest. Among the deep-learning methods, SteganoGAN (30.5 dB) is the only system that produces lower PSNR, due to its explicit deprioritisation of imperceptibility. All other methods that report PSNR figures fall in the 36–40 dB range, 3–7 dB above DASRS. TrustMark Q, at 43–45 dB, sets the ceiling against which DASRS must be contextualised.

The same DASRS configuration evaluated at higher resolutions (not the primary focus of this work) achieves approximately 37.64 dB at 1024×768 and 43.74 dB at 1920×1080, confirming that the imperceptibility deficit is a resolution-specific phenomenon that resolves at practical deployment scales ($\geq$1024 px wide).

Imperceptibility–Robustness Trade-off: Summary

Across the primary 512×512 SIPI evaluation, DASRS accepts a 4.63 dB PSNR reduction relative to the DCT baseline in exchange for a 32.82 percentage-point gain in mean PRR across the full 43-attack battery. This trade-off is the core empirical finding of the thesis: multi-domain redundancy purchases decisive robustness at a quantifiable and manageable imperceptibility cost.

Table 4.24: Imperceptibility comparison: cover-to-stego PSNR (dB). DASRS figures are empirical means over the USC-SIPI dataset; all other figures are reproduced from published papers and are indicative for their respective payloads and resolutions. Hybrid methods are discussed in Section 4.3.13. †Image-hiding task; ‡Generative method; N/R = not reported.

Method	**Year**	**Type**	**PSNR (dB)**
Classical baselines (this work)			
DCT baseline	—	Classical / single-domain	37.68
DWT baseline	—	Classical / single-domain	43.27
Deep-learning methods (published figures)			
HiDDeN [25]	2018	CNN enc-dec	36.1
SteganoGAN [29]	2019	GAN-based	30.5
UDH [44]	2020	CNN enc-dec	39.8
ReDMark [30]	2021	Diffusion res.	36.2
MBRS [1]	2021	Mixed-JPEG	38.1
HiNet [35]	2021	INN†	37.8
LIDS [34]	2022	Freq-constrained CNN	N/R
Mareen et al. [37]	2023	Geometric-robust CNN	N/R
PRIS [36]	2023	Robust INN	N/R
TrustMark [39]	2023	GAN spatio-spectral	43–45
Zhou et al. [38]	2024	Adaptive MoE	>39.8
LDStega [42]	2023	Latent diffusion‡	N/R
Hybrid methods (published figures)			
CNN-DCT [**?**]	2024	Hybrid (CNN+DCT)	42.5
DCT–GAN [**?**]	2025	Hybrid (DCT+GAN)	58.27
RT–ILWT [**?**]	2025	Hybrid (RT+ILWT)	56.22
StegTransX [**?**]	2025	Hybrid (CNN+JPEG)	>40
Proposed method (this work)			
DASRS	**2025**	**Multi-domain classical**	**33.05 (512×512 SIPI mean)**

Table 4.25: trade-off between DASRS and the DCT baseline.

Resolution	**DASRS PSNR**	**DCT PSNR**	**ΔPSNR vs DCT**	**ΔMean PRR vs DC**
512 × 512 (SIPI, primary)	33.05 dB	37.68 dB	−4.63 dB	+32.82%
1024 × 768 (indicative)	37.64 dB	52.08 dB	−14.44 dB	—
1920 × 1080 (indicative)	43.74 dB	57.38 dB	−13.64 dB	—

Conclusion and perspectives

In conclusion, this thesis has presented DASRS (Damage Aware Self-Recovering Steganography), a robust steganographic framework that fundamentally shifts the design focus from concealment-centric embedding to payload-oriented survivability. By treating the hidden message as a structured, redundant data object rather than a raw bit sequence, DASRS achieves measurable robustness advantages over classical single-domain methods and competitive performance relative to state-of-the-art deep learning approaches, while providing a partial-recovery guarantee that neither paradigm currently offers.

4.5 Summary of Research Objectives and Achievements

The primary objective of this research was to design a robust steganographic framework that prioritises payload survivability over mere concealment, enabling partial recovery of hidden information under adverse image degradation. The specific objectives stated in Chapter 1 are revisited below with an assessment of their achievement.

Objective 1: Design a robust steganographic framework. This objective was achieved through the development of DASRS, a multi-layer framework comprising three independent architectural layers: the structured payload layer, the multi-domain embedding layer, and the damage-aware extraction layer. The framework was fully implemented in Python and evaluated against 43 attacks across the cover images. The experimental results demonstrate that DASRS achieves a mean Payload Recovery Rate (PRR) of 85.02% across the full attack battery, substantially exceeding the single-domain baselines (DCT baseline: 57.81%, DWT baseline: 46.91%).

Objective 2: Structure the secret payload into interconnected fragments. This objective was achieved through the design of a six-data-fragment plus one-XOR-parity-fragment architecture. Each fragment is protected by Reed–Solomon error correction and carries a CRC-16 integrity checksum. The XOR parity mechanism enables exact algebraic reconstruction of any single lost data fragment from the five surviving fragments and the parity fragment, as proven in Equation (2.3) and validated experimentally in Chapter 4.

Objective 3: Distribute message fragments across stable image regions and multiple transform domains. This objective was achieved through the four-domain embedding strategy: Spatial QIM on the Y channel, DCT comparison on the Y channel, DCT comparison on the Cr channel, and Haar DWT level-2 QIM on the Cb channel. Zone partitioning guarantees exclusive embedding regions for each fragment, preventing inter-fragment collision. The domain cascade extraction order (Cb/DWT $\rightarrow$ Cr/DCT $\rightarrow$ Y/Spatial $\rightarrow$ Y/DCT) was empirically validated in Section 4.3.12.

Objective 4: Develop an extraction and reconstruction strategy. This objective was achieved through the triple-validation cascade extraction mechanism (Reed–Solomon decode $\rightarrow$ CRC-16 check $\rightarrow$ Fragment ID verification) and the automatic XOR reconstruction trigger. The system returns both the recovered message and a quantitative PRR score, enabling graded assessment of damage extent rather than binary pass/fail reporting.

Objective 5: Evaluate the framework using standard metrics. This objective was achieved through comprehensive experimental evaluation using PSNR and SSIM for imperceptibility, BER and NC for bit-level fidelity, and the novel PRR metric for partial recovery quantification. The evaluation protocol comprised 33 single attacks and 10 compound attacks across nine distortion

categories,with all experiments conducted on eleven 512×512 images from the USC-SIPI dataset; PSNR figures at higher resolutions are indicative extrapolations at 1024×768, and 1920×1080.

4.6 Key Contributions

The main contributions of this thesis are:

1. **A payload-survivability-oriented design paradigm.** DASRS shifts the steganographic design focus from concealment-centric embedding to structural resilience, treating the hidden message as a redundant data object rather than a raw bit sequence.

2. **A two-level error protection architecture.** The combination of intra-fragment Reed–Solomon coding (correcting up to 8 byte errors per fragment) and inter-fragment XOR parity (enabling exact single-fragment reconstruction) provides complementary protection against bit-level corruption and fragment-level erasure.

3. **A multi-domain cascade extraction mechanism.** The four-domain embedding with priority-ordered cascade extraction ensures that fragment recovery succeeds whenever at least one domain survives, with the extraction log providing direct observability of domain-level robustness.

4. **The Payload Recovery Rate (PRR) metric.** PRR quantifies partial recovery outcomes that binary pass/fail metrics cannot express, enabling rigorous comparison of robustness across attack scenarios and system configurations.

4.7 Future Work

Several directions for future research are proposed:

Advanced erasure coding. Replacing the single-XOR parity scheme with a Reed–Solomon code over the fragment sequence would enable recovery from any two simultaneous fragment losses, at the cost of increased parity overhead. A systematic trade-off analysis between redundancy overhead and recovery guarantee is needed.

Real-world channel testing. End-to-end validation through actual social media and messaging platforms (Table 1.2) would directly test the practical relevance of the framework. This requires automation of the upload-download-extraction pipeline and collection of platform-specific distortion statistics.

Metadata self-embedding. Embedding the structural parameters currently stored in the external metadata dictionary as a compact header within the most robust domain (e.g., Cb/DWT level-2) would eliminate the external dependency. The header would need to be recoverable under the same attack conditions as the payload itself.

Steganalysis resistance evaluation. The detectability–robustness trade-off was discussed in Chapter 1 but not empirically evaluated. Testing DASRS against modern deep-learning steganalysers (e.g., SRNet, XuNet) would quantify the statistical footprint introduced by multi-domain embedding and inform embedding strength calibration.

4.7.1 Framework Generalisability

The experimental results validate a specific four-domain instantiation of DASRS across a diverse eleven-image dataset, but the structural guarantees — payload fragmentation, parity-based reconstruction, and damage-aware cascade extraction — are architectural properties of the framework itself, independent of the particular domain configuration used. Any set of structurally independent transform domains satisfying the mutual decorrelation requirement can replace the current

configuration without altering the recovery semantics. The limitations identified above are therefore addressable through targeted domain substitution rather than fundamental redesign, making the framework extensible to threat models beyond those evaluated in this work.

4.8 Closing Remarks

This thesis has presented DASRS, a steganographic framework built on the principle that the survivability of hidden data depends primarily on how that data is structured and distributed, rather than on the choice of any single embedding technique. By combining multi-domain redundant embedding, structured payload fragmentation with XOR parity, and damage-aware cascade extraction, DASRS achieves measurable robustness advantages over single-domain classical baselines and competitive performance relative to state-of-the-art deep learning methods, while providing a partial-recovery guarantee that neither approach currently offers. The framework's modular architecture and interpretable design parameters make it a practical and extensible foundation for robust covert communication in hostile and uncontrolled digital channels.

Bibliography

[1] 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.

[2] Eric Wengrowski and Kristin Dana. Light field messaging with deep photographic steganography. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1515–1524, 2019.

[3] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding — a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[4] 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`.

[5] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.

[6] John G. Proakis and Masoud Salehi. *Digital Communications*. McGraw-Hill, New York, NY, USA, 5th edition, 2008.

[7] Jessica Fridrich and Jan Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[8] Niels Provos and Peter Honeyman. Hide and seek: An introduction to steganography. In *IEEE Security & Privacy*, pages 32–44, 2003.

[9] Neil F. Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *IEEE Computer*, 31(2):26–34, 1998.

[10] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2nd edition, 2007.

[11] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[12] 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.

[13] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, 2000.

[14] J. J. K. Ó Ruanaidh and T. Pun. Rotation, scale and translation invariant digital image watermarking. In *IEEE International Conference on Image Processing*, 1998.

[15] Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, pages 529–541, 1981.

[16] V. Solachidis and I. Pitas. Circularly symmetric watermark embedding in 2-d dft domain. *IEEE Transactions on Image Processing*, 10(11):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] 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.

[19] V. Solachidis and I. Pitas. Circularly symmetric watermark embedding in 2-d fourier domain. *IEEE Transactions on Image Processing*, pages 1741–1753, 2004.

[20] Neil F. Johnson and Sushil Jajodia. Steganography: Seeing the unseen. In *IEEE Computer*, pages 26–34, 1998.

[21] Partha Chowdhuri, Pabitra Pal, and Tapas Si. A novel steganographic technique for medical image using SVM and IWT. *Multimedia Tools and Applications*, 82(13):20497–20516, January 2023.

[22] 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.

[23] Gregory K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[24] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, *Advances in Cryptology: Proceedings of CRYPTO 83*, pages 51–67. Plenum Press, 1984.

[25] 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.

[26] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[27] 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.

[28] Andreas Westfeld. F5 — a steganographic algorithm: High capacity despite better steganalysis. In *International Workshop on Information Hiding*, pages 289–302. Springer, 2001.

[29] 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.

[30] Souvik Das, Chia-Yu Lin, and Hung-Yu Wei. ReDMark: Framework for residual diffusion watermarking based on deep neural networks. *Expert Systems with Applications*, 166:114085, 2021.

[31] 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.

[32] Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017.

[33] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[34] Huajie Chen, Tianqing Zhu, Yuan Zhao, Bo Liu, Xin Yu, and Wanlei Zhou. Low-frequency image deep steganography: Manipulate the frequency distribution to hide secrets with tenacious robustness. *arXiv preprint arXiv:2303.13713*, 2023.

[35] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. HiNet: Deep image hiding by invertible network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2021.

[36] Hang Yang, Yitian Xu, Xuhua Liu, and Xiaodong Ma. PRIS: Practical robust invertible network for image steganography. *Engineering Applications of Artificial Intelligence*, 2024.

[37] Hannes Mareen, Lucas Antchougov, Glenn Van Wallendael, and Peter Lambert. Blind deep-learning-based image watermarking robust against geometric transformations. In *arXiv preprint arXiv:2402.09062*, 2024.

[38] Yiqiao Zhou, Na Wang, Xiaolong Hong, Yanchun Peng, and Shuo Shao. Deep learning-based image steganography with latent space embedding and smart decoder selection. *Entropy*, 27(12):1223, 2025.

[39] Tu Bui, Shruti Agarwal, and John Collomosse. TrustMark: Universal watermarking for arbitrary resolution images. In *arXiv preprint arXiv:2311.18297*, 2023.

[40] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024.

[41] Daegyu Kim, Hyun Lee, Jaeho Park, et al. Diffusion-stego: Training-free diffusion generative steganography via message projection, 2023.

[42] Yinyin Peng, Yaofei Wang, Donghui Hu, Kejiang Chen, Xianjin Rong, and Weiming Zhang. LDStega: Practical and robust generative image steganography based on latent diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pages 3001–3009. ACM, 2024.

[43] Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. RoSteALS: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

[44] 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.

[45] S. Ahmad et al. Deep learning-based image steganography over cloud. *ACM Computing Surveys*, 2024. DOI: 10.1007/s42979-024-02756-x.

[46] R. Al-Rawashdeh et al. Robust image steganography approach based on edge detection combined with CNN algorithm. *IEEE Access*, 2025. In press.

[47] Kaleem Razzaq Malik et al. A hybrid steganography framework using DCT and GAN for secure data communication in the big data era. *Scientific Reports*, 15:9854, 2025.

[48] A. Khan and Y. Yildiz. Wavelet-based fusion for image steganography using deep convolutional neural networks. *Electronics*, 14(14):2758, 2025.

[49] A reversible and robust hybrid image steganography framework using radon transform and integer lifting wavelet transform. *Scientific Reports*, 15:98539, 2025.

[50] Xintao Duan, Zhao Wang, Sen Li, and Chuan Qin. StegTransX: A lightweight deep steganography method for high-capacity hiding and JPEG compression resistance. *Information Sciences*, page 122264, 2025.

[51] David A. Patterson, Garth Gibson, and Randy H. Katz. A case for redundant arrays of inexpensive disks (RAID). *ACM SIGMOD Record*, 17(3):109–116, 1988.

[52] Shu Lin and Daniel J. Costello. *Error Control Coding*. Pearson Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2004.

[53] W. Wesley Peterson and Edward J. Weldon. *Error-Correcting Codes*. MIT Press, 2nd edition, 1972.

[54] Gregory K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[55] 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.

[56] Andrew B. Watson. DCT quantization matrices visually optimized for individual images. In *Proceedings of SPIE Human Vision, Visual Processing, and Digital Display IV*, volume 1913, pages 202–216, 1993.

[57] Alfred G. Weber. The USC-SIPI image database: Version 5. Technical Report 315, Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, 1997. Available: `http://sipi.usc.edu/database/`.