Does the Source of Carrier Image Affect Steganographic Detectability?

A Comparative Study of Real vs. ML-Generated Image Steganography

Nico | Nikolas | Abdul | Daria | Jimena | David

Department of Advanced Computing Sciences
Maastricht University

Project 2.2 | February 2026

Agenda

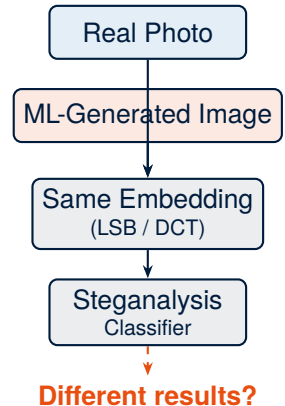
1. The Big Idea
2. Why This Study Matters
3. Research Questions
4. Experimental Setup
5. Hypotheses
6. Literature Coverage
7. Pros & Cons
8. Feasibility & Timeline
9. Risk Mitigation
10. Discussion

What Are We Proposing?

Image steganography hides secret data inside image files.

Its detectability depends on the **statistical properties** of the carrier image.

ML-generated images (Stable Diffusion, StyleGAN3, DALL-E) have fundamentally different statistical properties than real photographs.



Core Question

Does the **origin** of the carrier image (real vs. ML-generated) change how easy it is to **detect** hidden messages?

Why Should We Care?

1. Security Implications

- If ML images are harder to steganalyze $\rightarrow$ adversaries exploit synthetic carriers
- If easier to detect $\rightarrow$ new forensic opportunities

2. Timely & Novel

- AI images are everywhere (social media, news, art)
- **No systematic study** compares real vs. ML images as steganographic carriers under controlled conditions
- De et al. (2022) touched this space but did not do a controlled comparison

3. Scientifically Interesting

- Generative models impose learned statistical regularities (GAN spectral peaks, diffusion noise patterns)
- How do these interact with steganographic distortion?
- Tests cross-domain classifier generalization

4. Practical Scope

- Well-defined, testable hypotheses
- Clear experimental pipeline
- Open-source tools + our own hardware
- Quantitative results (AUC, accuracy, EER)

Research Questions

Primary Research Question

Does the source of the carrier image (real human-photographed vs. ML-generated) affect the detectability of image steganography when using identical embedding methods and payload sizes?

RQ1: Payload

How does payload size influence detectability across real and ML-generated images?

RQ2: Method

Do LSB (spatial) and DCT (freq-domain) behave differently by carrier origin?

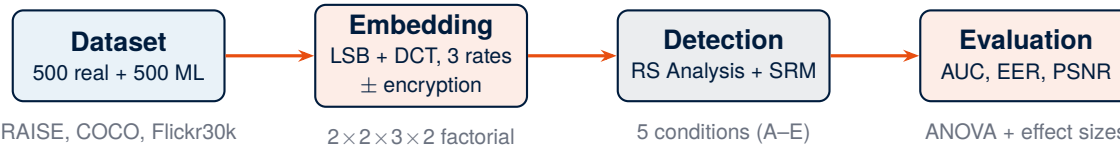
RQ3: Encryption

Does encrypting the payload before embedding affect detectability?

RQ4: Cross-Domain

Do classifiers trained on real images generalize to ML-generated images, and vice versa?

Experimental Pipeline



4 Factors

Carrier × Method × Payload (3) × Detector

Matched Pairs

Real photos matched to ML images by semantic content

5 Train/Test Splits

Within-domain, cross-domain, and mixed

Datasets

Real Images (500 photos)

- **RAISE**: High-quality RAW photos from DSLR cameras, diverse scenes
- **COCO / Flickr30k**: Natural photographs, diverse content
- Normalized: 512×512 px, RGB, 8-bit PNG

ML-Generated Images (500 images)

- **Stable Diffusion**: Latent diffusion, photorealistic text-to-image
- **StyleGAN3**: GAN-based, well-characterized spectral artifacts
- Same 512×512, RGB, 8-bit PNG format

Matching protocol: ML images generated from same semantic prompts as real image content · Identical format specs · Consistent mean luminance

Embedding, Payload Rates & Encryption

LSB Substitution (spatial domain)

- Replace k least significant bits of pixel values
- Pseudorandom pixel selection across RGB channels
- High capacity, lower robustness

DCT-Based Embedding (frequency domain)

- 8×8 pixel blocks, QIM on mid-frequency DCT coefficients
- Mirrors JPEG-domain steganography (F5 / JSteg)
- Better imperceptibility, lower

Level	LSB	DCT
Low	$k = 1$, 25% px	10% coeff.
Medium	$k = 1$, 50% px	25% coeff.
High	$k = 2$, 50% px	50% coeff.

bpp = bits per pixel

Cross-Domain Conditions

	Train on	Test on	What it tells us
A	Real	Real	Baseline: standard steganalysis performance
B	ML-gen	ML-gen	Is ML imagery inherently easier/harder to steganalyze?
C	Real	ML-gen	Can a "real-trained" detector catch ML-embedded stego?
D	ML-gen	Real	Can an "ML-trained" detector catch real-embedded stego?
E	Mixed	Both	Does mixed training solve the domain gap?

Key Insight

Conditions **C** and **D** directly test **RQ4** (cross-domain generalization). Comparing **A** vs. **B** addresses the **primary RQ**.

Hypotheses

H1 – Distributional Difference ML images will show *different* detectability due to learned statistical regularities (GAN spectral peaks, diffusion noise patterns, smoother textures).

H2 – Payload Divergence The gap between carrier types *widens* at higher payloads. At low rates, both may be equally hard to detect.

H3 – Method Sensitivity DCT embedding will be *more* sensitive to carrier origin than LSB, because DCT coefficients are directly shaped by the generative process.

H4 – Cross-Domain Drop Classifiers will lose **10–25% AUC** when tested across domains, mirroring image deepfake detection findings.

H5 – Asymmetric Transfer Real→ML transfer (C) will perform *worse* than ML→Real (D), because ML images' distinctive statistical profile masks learned artifacts.

H6 – Encryption Effect Payload encryption will not significantly affect detectability, since AES output is statistically similar to a pseudorandom bitstream.

Literature Coverage

Image Steganography Surveys

Petitcolas et al. (1999) – Hussain et al. (2018)

Image Steganalysis

Westfeld & Pfitzmann (1999) – Chi-square attack

Fridrich et al. (2001) – RS Analysis

Fridrich & Kodovský (2012) – Rich Models (SRM)

Embedding Theory

Holub et al. (2014) – HILL/S-UNIWARD

Westfeld (2001) – F5 algorithm

ML Image Generation

Rombach et al. (2022) – Stable Diffusion

Karras et al. (2021) – StyleGAN3

OpenAI (2023) – DALL-E 3

AI + Steganography (The Gap)

De et al. (2022) – AI image stego (closest prior work)

Duan et al. (2020) – coverless GAN stego

Deepfake Detection & Datasets

Wang et al. (2020) – CNN-generated image detection

Corvi et al. (2023) – diffusion model detection

RAISE, COCO, Flickr30k datasets

Pros & Cons

Pros

- **Clear research gap** – no prior systematic controlled study
- **Well-defined** – testable hypotheses, quantitative metrics
- **Runs locally** – Stable Diffusion + StyleGAN3 on M4 Pro, no cloud GPUs
- **Modular pipeline** – independent stages, parallelizable
- **Publishable angle** – timely intersection of two active fields
- **Better tooling than audio** –

Cons / Risks

- **6–7 weeks is tight** – disciplined timeline needed
- **Null result risk** – carrier origin may show no effect
- **Dataset size** – 500 per carrier type; covers the main experimental conditions
- **Scope creep** – tempting to add more generation models
- **DCT complexity** – JPEG-domain embedding requires careful block-level implementation

Feasibility & Timeline

Hardware M4 Pro MacBooks (18–36 GB unified memory)

Image generation

- Stable Diffusion: ∼3–5 h for 250 images (MPS)
- StyleGAN3: ∼2–3 h for 250 images
- Real images: download from RAISE/COCO

Detection (CPU-only)

- RS Analysis & chi-square: seconds per image, no training
- SRM + FLD: ∼30 min total (scikit-learn, CPU)

Week	Activity
1	Data collection & ML image generation
2	Stego implementation (LSB + DCT ± A
3–4	Classifier training (all conditions)
5	Cross-domain experiments & analysis
6	Paper writing & visualization
7	Buffer / revision / submission

Parallelization

Wk 1–2: split data gen & embedding
Wk 3–4: each member trains subset
Wk 5–6: analysis & writing in parallel

Risk Mitigation

Risk	Mitigation
Null result	A null result *is* a valid finding – shows existing steganalysis generalizes well across carrier origins. Frame accordingly.
Dataset too small	Data augmentation (flips, crops, color jitter on cover images before embedding); target 500 total if time allows.
Semantic matching imperfect	Define matching as same semantic category (not same scene); document as a design choice.
DCT implementation	Start with LSB (simpler); use reference implementations (OpenStego); DCT as secondary.
Scope creep	Freeze design after Week 2; additional generation models go into “future work.”

Discussion

1. **Dataset size:** 500 per carrier type (250 per model) – extended from 300 to improve statistical power

2. **Generation models:** Stable Diffusion + StyleGAN3 sufficient, or add DALL-E 3 as a third (API-based)?

3. **Division of labor:** Proposed split –
   - *Data team* (2): dataset collection, ML image generation, normalization
   - *Stego team* (2): LSB + DCT implementation, AES encryption, embedding pipeline
   - *Classification team* (2): classifier training, evaluation, statistics

4. **Matching definition:** Same scene vs. same semantic category – which is more defensible?

5. **Should we proceed with this topic?**

Thank you

Questions & Discussion

Full proposal PDF and reference review available for detailed reading.