**PROJECT TITLE**

# Dual-Modality Steganography: Enhanced LSB for Text-in-Image and Optimized Frame Selection for Video-in-Video Embedding


## ABSTRACT

With the increasing demand for digital privacy, secure and undetectable data transmission has become a growing necessity across various fields. Existing image steganography methods like Least Significant Bit (LSB) substitution face a trade off between payload capacity and detectability, with the neural network based steganalysis achieving up to 96.78% accuracy when the payload s reach 0.5 bits per pixel. This project proposes an enhanced LSB approach that allows users to hide text within an image and embed video within another video while minimizing visual distortion and reducing detection risk. This proposed approach aims to outperform traditional LSB (which typically achieves SSIM < 0.90 for 5,000-word payloads) and reduce detectability by 60% compared to state-of-the-art neural network steganalysis. The tool will target the ability to embed more than 5,000-word text into PNG images while maintaining SSIM ≥0.95 and reducing detectability to at least 15%. By using Python to build the interface design and C++ to enhance the processing performance, the tool ensures both usability and speed without requiring high-end hardware. The decoding accuracy is expected to achieve a text or video retrieval error rate of less than 1% across 100 test samples. By addressing the technical limitations, this tool will be a practical, accessible, and efficient solution for secure content hiding across multiple real world applications.


## PROBLEM

A major challenge in current information security is ensuring the secure and hidden data transmission. Encryption only protects the content of a message while steganography hides the presence of the data within media. However, the existing steganography tools face technical limitations, computational inefficiency, and limited real world applicability.

1. **Technical Limitation in Traditional Image Steganography**
   Most image steganography tools rely on basic techniques, such as Least Significant Bit (LSB) substitution as it is easy to implement. However, as the embedded payload increases, specifically between 0.1 to 0.5 bits per pixel, the detectability via neural network-based steganalysis rises significantly which reaches up to 96.78% accuracy (Julián D. & Diego J., 2021). It also reduces the image quality because of the changes in brightness and their pixel patterns. Furthermore, 60% of existing tools support only BMP and JPEG formats which reduces the suitability for different operational environments (Richard et al., 2024).

2. **Challenges in Video Steganography**
   Although video provides a higher capacity compared to the images, many existing steganography techniques are not optimized for high resolution media. Methods like frame-by-frame embedding increases computational load by 40% (O. Oleg et al.,

2020) and causes 15% synchronization errors in 1080p videos which make the real-time use impractical.

3. **Challenges with Modern AI-based Techniques**
   Recent research has introduced deep learning based steganography that selectively hides inter-frame residuals using neural networks to improve efficiency but still requires high processing power and memory (Fariha & G. R., 2019). Besides, hiding important objects like faces or backgrounds (Xueying et al., 2024), which have demonstrated an enhanced ability to conceal data and avoid detection. However, these approaches are difficult to apply in the real world applications due to it requiring more than 8GB RAM and taking approximately twice longer processing time. This high resource demand limits their usability on standard devices.

**RESEARCH QUESTIONS**

**R1:** Can adaptive LSB (allocating 1 to 4 bits per pixel based on image texture) enable hiding 5,000-word text in PNGs with PSNR ≥ 35dB and steganalysis detectability < 10% using the J-UNIWARD steganalyzer?

**R2:** Will selective frame encoding (prioritizing static frames for embedding) reduce video synchronization errors to < 0.5 frames and computational time by 50% vs. frame-by-frame methods?

**R3:** Can a Python-C++ hybrid tool (Python for UI, C++ for parallel processing) achieve text-in-image embedding speeds ≥ 2 MB/s and video-in-video speeds ≥ 10 fps on standard laptops (defined as Intel Core i5-1135G7, 8GB RAM, and integrated GPU)?

**HYPOTHESIS**

**H1:** Enhanced LSB with adaptive bit allocation and AES-256 encryption will increase text payload capacity by 50% (vs. basic LSB) while maintaining SSIM ≥ 0.95 and reducing detectability to < 15%.

**H2:** Selective frame encoding (embedding in 30% of static frames) will reduce video quality degradation (VMAF ≥ 90) and synchronization errors to < 1 frame, with 40% lower CPU usage.

**H3:** A Python and C++ hybrid tool will achieve text-in-image embedding speeds ≥ 2 MB/s and video-in-video speeds ≥ 10 fps on standard laptops (Intel Core i5, 8GB RAM) to ensure real-world usability without deep learning.

**OBJECTIVE OF THE STUDY**

**O1:** To implement an enhanced image steganography module that is able to hide 5,000-word text in PNGs with SSIM ≥ 0.95 and steganalysis detectability < 10%.

**O2:** To develop a video steganography module that is able to embed 5 minutes 720p videos into 10 minutes 1080p videos with synchronization errors < 0.5 frames and VMAF ≥ 90.

**O3:** To evaluate the performance, robustness, and usability of the proposed steganography tool across various formats by using both Python and C++.

**SOLUTION**

This project will develop a steganography tool that is able to hide text inside an image, and embed video into another video. The tool will consist of two main modules:

- **Image steganography module with hidden text**
  This module enables users to hide text with flexible format into an image file using enhanced Least Significant Bit (LSB) techniques. The embedding process will use the Canny edge detection algorithm to allocate 1 bit per pixel in high-texture areas and 3 to 4 bits in smooth areas. To secure the payload, text will be encrypted using AES-256 encryption with keys derived.The keys will be derived using PBKDF2 (Password-Based Key Derivation Function 2) with SHA-256 hashing and a random salt (per NIST guidelines) to enhance the resistance to brute-force attacks.

- **Video steganography module with hidden video**
  This module enables users to embed video into another video using optimized frame selection and lightweight compression strategies to preserve playback quality. It will apply optical flow to identify static frames (motion vector <5 pixels) for 70% of embedding, with dynamic frames using H.265 compression to reduce payload size by 30%.

It will be a desktop based tool that allows users to select a cover image or video, input or upload the secret text or video to hide, encode and decode data without needing complex AI, and optional encryption by setting password or time control for decode. The tool improves the traditional LSB methods by adding format flexibility and optional encryption. This helps to reduce the risk of detection by unauthorized parties and minimizes visual distortion in image steganography. It also avoids the high complexity of the modern AI based frameworks that is commonly used in video steganography. By using Python and C++, it offers speed and accessibility that is suitable for developers and general users with standard computing devices.

To ensure the reliability, the decoding process will be tested across 100 media samples consisting of 50 images from the DIV2K dataset, 50 from the CLIC dataset, and 20 short videos from the UCF101 action recognition dataset. For image steganography, the accuracy will be evaluated using bit error rate (BER) between the original and extracted text. For video steganography, reliability will be evaluated by calculating the frame loss rate. The overall performance of the tool shall maintain an error rate below 1%.

Critical functions of the steganography tool:

- Hiding text with flexible file format into an image file like .PNG with minimal visual degradation.
- Embedding a smaller video file into a larger video file while maintaining the video's playback quality.

- Accurately decode and retrieve the hidden text or video from the media.
- Optional encryption by setting password or time control during the encode process. When users wish to decode the media, they have to enter the correct password or after the pre-set time.

**TARGET MARKET**

1. **Photographers and Digital Artists**
   Potential users include photographers and digital artists, who could embed invisible watermarks or proof of ownership into their images—if the module meets its design objectives (SSIM $\geq 0.95$). It helps to protect their intellectual property in case their image is stolen or reused. Compared with the visible watermarks, the hidden data cannot be easily removed or edited.

2. **Educators, Students, and Academic Institutions**
   In the educational field, exam papers always have the risk of being intercepted by unauthorized parties during file transfer between different branches. This tool helps to mitigate the risk by allowing educators to embed confidential materials into an image file and optional time control feature for decode. In this way, the media seems less attractive and cannot be decoded before the pre-set time.

3. **Journalists, Whistleblowers, and Activists**
   The journalists or activists often work in high risk regions. By using this tool, they can embed sensitive documents or footage into common images or videos to minimize the exposure of original media. Since it operates entirely offline and does not rely on the cloud, the process is secure and private. It helps to transmit information secretly without drawing attention.

**COMPETITION / CONTRIBUTION**

This project introduces a dual-function steganography tool that supports both hidden text in image and hidden video in video embedding, which is a combination that is rarely found in existing tools. It is built with a hybrid of Python for user interface and flexibility, and C++ for the performance handling. Furthermore, this project supports widely used media formats like PNG and MP4, and can operate completely offline. Therefore, it is more secure and enhances data privacy. Practical applications include enabling photographers and digital artists to embed invisible watermarks via text-in-image steganography, allowing educators to securely distribute confidential materials with optional timed decoding, and supporting journalists or activists in high-risk regions to discreetly transmit sensitive footage via video-in-video steganography.

In terms of contribution:

- **Stakeholders** can have a reliable and low-cost solution for secure data handling.
- **Communities** gain access to an open source and extendable platform for future research or development.
- **End users** from multiple fields like digital media, education, and journalism can securely hide and transfer content without drawing attention.

**MILESTONES**

**Proposed Development Model / Research Method** : Waterfall based development which includes requirement analysis, design, implementation, and testing.

**Project Schedule:**

| ACTIVITIES | EXPECTED OUTCOME | COMPLETATION DATE |
|---|---|---|
| Proposal Submission | All components of the full project proposal, including the required forms and documentation will be completed and approved by the supervisor | 17/8/2025 |
| Chapter 1: Introduction Submission | The introduction chapter which includes the project background, scope, aim, and justification will be prepared and submitted. | 22/8/2025 |
| Chapter 2: Background Research Submission | A detailed literature review and background study that support the project's relevance and conceptual foundation will be completed and submitted. | 8/9/2025 |
| Chapter 3: Methodology & Requirement Analysis Submission | The chapter outlining the research methodology and the system requirements analysis will be completed and submitted. | 22/9/2025 |
| Image Module Prototype Testing | Complete prototype development for adaptive LSB; prepare 50 test images (DIV2K/CLIC) and validation protocol to measure SSIM $\geq 0.95$ | 30/10/2025 |
| Chapter 4: System Design Submission | The system architecture, steganography module design, encryption module design, and relevant technical diagrams will be submitted. | 21/11/2025 |
| Project I Portfolio Submission | A complete project portfolio including Chapter 1 until 4 along with all supporting documentation will be compiled and submitted in accordance with the institution's guidelines. | 12/12/2025 |
| Video Module Optimization | The system reduces the synchronization errors to less than 0.5 frames via frame | 15/1/2026 |

| | | |
|---|---|---|
| | selection tuning on the video steganography tool. | |
| Initial System Preview | A functional prototype or initial system preview will be demonstrated to the supervisor to signify the transition to Project II. | 20/2/2026 |
| Final System Testing | Final testing and validation will be carried out under the supervision of both the supervisor and the assigned moderator. | 6/3/2026 |
| Draft FYP Report | An initial draft of the complete FYP report that includes all chapters and system documentation will be done and submitted for review. | 3/4/2026 |
| Final FYP Report Submission | The finalised FYP report that covers all required revisions and additional deliverables will be submitted in accordance with the official submission requirements. | 13/4/2026 |

**REFERENCES**


Dilara, D., & Selda, S. (2024). Enhancing Steganography in 256×256 Colored Images with U-Net: A Study on PSNR and SSIM Metrics with Variable-Sized Hidden Images. *Review of Computer Engineering Studies*, *11*(2), 13–29. https://doi.org/10.18280/rces.110202

Doaa A., D. A., & Mohmmed J., M. J. (2022). Comprehensive Survey of Multimedia Steganalysis: Techniques, Evaluations, and Trends in Future research. *Symmetry*, *14*(1), 117. https://doi.org/10.3390/sym14010117

Fariha, F., & G. R., G. (2019). Video Steganography using Convolutional Neural Network and Temporal Residual Method. *International Journal of Computer Applications*, *178*(46), 24–29. https://doi.org/10.5120/ijca2019919375

Julián D., J. D., & Diego J., D. J. (2021). LSB steganography detection in monochromatic still images using artificial neural networks. *Multimedia Tools and Applications*, *81*(1), 785–805. https://doi.org/10.1007/s11042-021-11527-2

Oleg, O., Anna, A., & Roman, R. (2020). Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions. *IEEE Access*, *8*, 166589–166611. https://doi.org/10.1109/access.2020.3022779

Richard, R., Michael, M., Frimpong, F., James Ben, J. B., & Kwame Ofosuhene, K. O. (2024). Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review. *PLoS ONE*, *19*(9), e0308807. https://doi.org/10.1371/journal.pone.0308807

Xueying, X., Xiaoxiao, X., Wanli, W., Zhenliang, Z., Qichao, Q., Zhenxing, Z., Sheng, S., & Xinpeng, X. (2024). From covert hiding to visual editing: robust generative video steganography. *arXiv preprint arXiv:2401.00652 (submitted to IEEE Transactions on Multimedia)*. https://doi.org/10.48550/arxiv.2401.00652