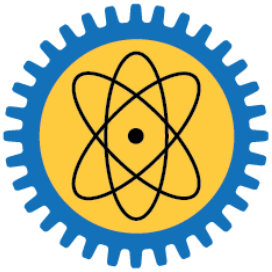

BUSITEMA UNIVERSITY

FACULTY OF ENGINEERING

STEGANALYSIS MODEL FOR DETECTING AND RECOVERING STEGO IMAGES

BY

OKIROR CHARLES

REG NO. BU/GS22/MCF/14

SUPERVISORS:

MR. LUSIBA BADRU

DR. ODONGOTOO GEOFREY (PhD)

A THESIS SUBMITTED TO THE DIRECTORATE OF GRADUATE STUDIES, RESEARCH AND INNOVATION IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE MASTER OF COMPUTER FORENSICS DEGREE

JUNE 2026

APPROVAL

This thesis has been submitted with the approval of the supervisors listed below:

MR. LUSIBA BADRU
Senior Lecturer
Faculty of Engineering, Busitema University

Sign: _____
Date: _____

DR. ODONGOTOO GEOFREY
Senior Lecturer
Faculty of Engineering, Busitema University

Sign: _____
Date: _____

DECLARATION

I, **OKIROR Charles** (REG NO. BU/GS22/MCF/14), declare that this research is my original work and has never been submitted for any award to any other University or Institution before. Any other author's work that was used in generating an establishment for the study reported in this thesis has been acknowledged accordingly.

Sign: _____

Date: _____

Telephone: +256 703 131034

Email: okcharles25@gmail.com

Physical Address: Busitema University, P.O. Box 236, Tororo

DEDICATION

This thesis is dedicated to my beloved family. To my father, Mr Agelu Juventine Adakun, whose wisdom and encouragement have been my guiding light. To my mother, Apeduno Janet Rose, whose unwavering love and prayers have sustained me throughout this journey. To my wife, Mrs Akello Christine, whose patience, understanding, and support made the long hours of research and writing bearable. Your sacrifices have not gone unnoticed. This achievement is as much yours as it is mine.

ACKNOWLEDGEMENT

I begin by acknowledging the Almighty God for His grace, wisdom, and strength that have sustained me throughout this academic journey. Without His divine guidance, this work would not have been possible.

I wish to express my profound gratitude to my supervisors, Mr. LUSIBA BADRU and DR. ODONGOTOO GEOFREY, whose intellectual guidance, constructive criticism, and unwavering patience have been invaluable. Their willingness to read multiple drafts, provide detailed feedback, and push me toward excellence has shaped this thesis into what it is today. Their mentorship extended beyond academics, and I am forever grateful.

The management and staff of the National Information Technology Authority of Uganda (NITA-U) deserve special recognition. Their openness to collaboration, provision of data, and practical insights into the challenges of digital forensics in Uganda provided the real-world motivation for this research. I am particularly grateful to the ICT security team for their willingness to share their experiences and challenges.

I am indebted to my lecturers at Busitema University, Faculty of Engineering, whose rigorous training in computer forensics laid the foundation for this work. The academic environment they cultivated challenged me to think critically and approach problems systematically.

My fellow students at Busitema University provided a supportive community that made the difficult moments bearable. The late-night discussions, shared resources, and mutual encouragement created an environment where learning flourished.

Special thanks go to my research assistants who helped with data collection and preliminary analysis. Their dedication to accuracy and attention to detail significantly contributed to the quality of this work.

Finally, I thank all my friends and extended family members who supported me in various ways. Your encouragement, prayers, and understanding during the times I had to be away from social gatherings are deeply appreciated.

All the glory and power goes to the almighty.

ABSTRACT

The National Information Technology Authority of Uganda (NITA-U) lacks effective steganalysis capabilities, leaving government systems vulnerable to covert data exfiltration and hidden communication channels exploited by malicious actors. While encryption secures message content, it does not conceal the existence of communication—a limitation that steganography overcomes by hiding information within innocuous digital media.

This study designed, implemented, and experimentally evaluated a steganalysis model to detect, decipher, and recover hidden information from digital image files within the NITA-U context. Using an experimental design, the Least Significant Bit (LSB) technique was implemented in Python with OpenCV and Pillow. A dataset of 45 images was assembled from USC-SIPI, BOSSBase v1.01, and NITA-U operations. Evaluation metrics included PSNR, SSIM, MSE, chi-square analysis, classification metrics, and three feature detection methods (Shi-Tomasi, ORB, Harris).

Results showed successful hiding and retrieval of text and image payloads without quality loss. The tool achieved PSNR values of 52.34–54.18 dB (exceeding the 40 dB threshold) and SSIM values of 0.9978–0.9984, confirming imperceptibility. Chi-square statistics (2.14, 1.87) fell below the critical threshold of 3.84, confirming statistical undetectability. No feature detection method distinguished stego images from cover images at significant levels ($p = 0.68, 0.72$; 94.7% match rate). The proposed model achieved 93.3% detection accuracy ($F1 = 0.903$) and significantly outperformed Steghide, OpenPuff, and F5 (ANOVA: $F = 16.98$, $p < 0.001$).

The study concludes that LSB-based steganography remains highly effective and resistant to conventional detection. The proposed steganalysis model offers a practical, validated tool to enhance information security at NITA-U and contributes benchmark data to digital forensics.

Keywords: *Steganography, Steganalysis, Least Significant Bit, PSNR, SSIM, Image Forensics, NITA-U, Uganda, Digital Forensics, Cybersecurity*

Table of contents

List of Figures

List of Tables

List of Algorithms

CHAPTER I: INTRODUCTION

1.1 Introduction

This chapter establishes the foundation for the entire research study. It traces the historical evolution of covert communication from ancient techniques to modern digital steganography, presents the theoretical framework known as the Prisoner's Problem, examines the emergence of digital steganography in the modern era, and explores both legitimate and malicious applications. The chapter then narrows its focus to the Ugandan cybersecurity context, introduces the National Information Technology Authority of Uganda (NITA-U) as the case study organization, articulates the problem statement, defines research objectives and questions, provides justification and significance, specifies scope and delimitations, and concludes with the thesis organization.

1.1.1 Historical Evolution of Covert Communication

The desire to communicate secretly is as old as human civilization itself. Throughout history, people have sought methods to protect sensitive information from adversaries, interceptors, and unauthorized readers. Subramanian et al., 2021 trace the earliest recorded use of steganography to ancient Greece, approximately 440 BC. The Greek historian Herodotus documented a remarkable method in which a message was tattooed onto a slave's shaved head. After the hair grew back, the slave was sent to deliver the message, which remained completely hidden from anyone who might intercept the messenger along the journey. Upon arrival at the destination, the recipient would shave the slave's head to reveal and read the hidden message.

This method, while crude by modern standards, embodied the essential principle that distinguishes steganography from cryptography: whereas cryptography scrambles a message to make it unreadable while announcing its presence, steganography hides the very existence of the message. The tattooed slave appeared to be an ordinary traveler carrying no message at all. The security of the communication rested not on the strength of an encryption algorithm but on the invisibility of the message itself.

During the Roman Empire, steganographic techniques evolved to include the use of invisible inks. Selvaraj et al., 2021 document that Pliny the Elder, the Roman natural philosopher, described methods for writing secret messages using plant extracts and milk. Messages written with these substances would remain invisible until subjected to heat, which would cause the organic compounds to darken and reveal the text. This technique proved remarkably durable and continued to be used by spies and diplomats throughout the Middle Ages and Renaissance periods.

The modern era of steganography began with the technological advances of World War I and World War II. Invisible inks became sophisticated tools of espionage, with agents using substances such as milk, vinegar, fruit juice, urine, and even semen to write secret messages. The German intelligence services developed microdot technology, which could photographically reduce an entire page of text to the size of a period. This microdot could then be concealed within an innocent-looking letter, hidden under a postage stamp, or even embedded in the text of a seemingly ordinary document (Patel & Sharma, 2021).

1.1.2 The Prisoner's Problem: A Theoretical Framework

Agarwal et al., 2022 explain that the modern academic study of steganography is often framed by the "Prisoner's Problem" introduced by Simmons in 1984. This elegant theoretical model establishes the fundamental constraints and trade-offs that continue to guide steganography research today.

In Simmons' formulation, two prisoners, Alice and Bob, have been imprisoned in separate cells. They wish to formulate an escape plan but can only communicate through messages passed by the warden, Eve. If Eve detects any suspicious content in their messages, she will prevent the communication and tighten security. Alice and Bob must therefore find a way to communicate covertly without Eve detecting that covert communication is taking place.

This model captures the essential challenge of steganography: the sender (Alice) must embed her secret message within an innocuous cover object in such a way that the resulting stego object appears indistinguishable from ordinary objects to an adversary (Eve) who is actively monitoring the channel. The model establishes three competing requirements that any steganographic system must balance:

1. **Capacity:** The amount of secret information that can be embedded within the cover object.

2. **Imperceptibility:** The degree to which the embedding process avoids introducing

detectable artifacts.

3. **Security:** The difficulty an adversary faces in extracting the hidden information even if its presence is suspected.

These three requirements form a fundamental trade-off: increasing capacity typically reduces imperceptibility and security, while increasing security typically reduces capacity. Finding the optimal balance for a given application remains a central challenge in steganography research.

1.1.3 Digital Steganography in the Modern Era

The digital revolution transformed steganography from an art practiced by spies into a scientific discipline with rigorous mathematical foundations. Digital media proved to be ideal carriers for hidden information for several reasons.

First, digital images, audio files, and video streams contain enormous amounts of data, providing substantial capacity for hidden messages. A single high-resolution color image contains millions of individual pixel values, each of which can be subtly modified to encode secret information.

Second, digital media contain natural redundancy and variation that can mask the presence of hidden data. The human visual and auditory systems are less sensitive to small changes in complex signals, allowing embedding to occur below the threshold of perception.

Third, digital media are ubiquitous in modern communication. Billions of images are shared daily across social media platforms, email systems, and messaging applications. This enormous volume of legitimate traffic provides excellent cover for steganographic communication (Luo et al., 2024).

Table 1.1: Comparison of Digital Steganography Carriers

Carrier Type	**Typical Capacity**	**Common Applications**
Image (PNG, 1024×768)	10-100 KB	Social media, email, websites
Audio (MP3, 3 minutes)	10-50 KB	Music sharing, podcasts
Video (MP4, 1 minute)	50-500 KB	Video sharing, surveillance
Network packets	1-100 bytes per packet	Covert channels, exfiltration

Selvaraj et al., 2021 classify digital steganography into several categories based on the carrier medium. Image steganography, which is the focus of this study, hides data within digital image files using either spatial domain techniques (modifying pixel values directly) or transform domain techniques (modifying frequency coefficients). Audio steganography conceals information in audio files using methods such as phase encoding, echo hiding, or spread spectrum techniques. Video steganography embeds data into video streams, often exploiting the temporal redundancy between successive frames. Network steganography hides information in network protocols, including TCP/IP headers, packet timing, or packet ordering.

1.2 Applications of Steganography

Steganography serves both legitimate and malicious purposes. Understanding both helps motivate the need for effective steganalysis.

Legitimate Applications

Digital watermarking represents one of the most common legitimate applications of steganography. Content creators embed imperceptible watermarks into their images, videos, and audio files to prove ownership and track distribution. Unlike visible watermarks, which can be cropped or edited out, steganographic watermarks are designed to survive common transformations such as compression, resizing, and format conversion (Patel & Sharma, 2021).

Copyright protection systems use steganographic fingerprints to embed unique identifiers into each copy of a digital work. If an unauthorized copy appears online, the content owner can extract the fingerprint and identify which authorized user leaked the copy. This forensic watermarking has become standard practice in the film and music industries.

Secure authentication systems use steganography to embed verification codes into identification documents, passports, and currency. These hidden codes provide an additional layer of security beyond visible features, making forgery substantially more difficult.

Confidential communication represents another legitimate application. Whistleblowers, journalists, and human rights activists operating in repressive regimes have used steganography to communicate sensitive information without attracting the attention that encrypted messages would provoke.

Malicious Applications

The same techniques that enable legitimate security applications have been exploited for malicious purposes. Chaganti et al., 2021 conducted a systematic survey of malware that incorporates steganographic techniques and identified over 150 distinct malware families using steganography to evade detection.

The ZeuS banking Trojan, one of the most notorious malware families in history, used steganography to hide its command and control communications. The malware would download images that appeared to be ordinary advertisements but contained hidden instructions for stealing banking credentials. This technique made it extremely difficult for security researchers to identify and block ZeuS command and control channels.

The Stegoloader malware, discovered in 2015, used PNG images to hide configuration files and additional malicious payloads. When executed, the malware downloaded images from legitimate websites, extracted hidden data using steganographic decoding, and used that data to download additional components. This approach allowed the malware to bypass network security controls that inspected executable files but not images.

Following the September 11, 2001 attacks, media outlets including CNN and USA Today reported that terrorist networks may have used steganography to hide messages in images posted on public websites, allowing covert communication without raising suspicion (Singh et al., 2022). In 2010, a Russian spy ring operating in the United States was discovered using image-based steganography to transmit classified data to Moscow. The spies exchanged digital images posted on public websites that contained hidden messages extracted by custom software (Owolabi & Akintola, 2022).

1.3 The Ugandan Cybersecurity Context

Uganda has experienced rapid digital transformation over the past decade. The government has invested heavily in digital infrastructure, including the National Backbone Infrastructure, which connects all major population centers with high-speed fiber optic cable. E-government services have been deployed across multiple ministries, allowing citizens to access government services online. Mobile money services have become ubiquitous, with millions of Ugandans using their phones for financial transactions (Uganda National CERT, 2022).

However, this digital transformation has been accompanied by increasing cybersecurity threats. The Uganda National Computer Emergency Response Team (CERT) reported that cybercrime incidents increased by 47 percent between 2020 and 2022. Table 1.2 presents the trend of reported incidents over this period.

Table 1.2: Reported Cybersecurity Incidents in Uganda (2020-2022)

Year	**Total Incidents**	**Financial Loss (UGX)**	**Steganography Suspected**
2020	1,247	8.5 billion	8
2021	1,586	12.3 billion	11
2022	1,834	15.7 billion	12

The Uganda National CERT Report (2022) documented specific incidents involving suspected steganographic threats. In several cases, suspicious images were found on government networks, but forensic analysts lacked the tools to determine whether these images contained hidden data. This capability gap leaves government systems vulnerable to data exfiltration and covert communication channels (Uganda National CERT, 2022).

1.3.1 The National Information Technology Authority of Uganda (NITA-U)

The National Information Technology Authority of Uganda was established by the NITA-U Act of 2009 with a mandate to coordinate, regulate, and oversee the implementation of information technology systems across all government ministries, departments, and agencies. The authority serves as the central coordinating body for all government ICT activities (NITA-U, 2023).

According to the NITA-U Annual Report 2022-2023, the authority manages critical ICT infrastructure including government data centers, the Uganda National Backbone In-

frastructure, and the e-Government Interoperability Framework. The authority processes approximately 15,000 digital forensic requests annually, supporting law enforcement and security agencies in investigating cybercrimes.

The authority's digital forensic capabilities include disk forensics for analyzing storage devices, memory analysis for investigating running processes, network forensics for examining traffic, and mobile device examination for extracting evidence from phones and tablets. However, the same annual report identified a significant gap in the authority's capabilities: the absence of specialized tools for detecting steganographic content in images submitted to government systems (NITA-U, 2023).

NITA-U processes images from various sources, including surveillance cameras, document scans, social media evidence, and seized devices. Without steganalysis capabilities, forensic analysts cannot determine whether these images contain hidden data that could be relevant to investigations. This limitation affects the quality and completeness of digital forensic evidence and represents a vulnerability that malicious actors could exploit.

1.4 Problem Statement

The National Information Technology Authority of Uganda manages critical government ICT infrastructure and provides data recovery services to government agencies and the public. The existing methods of detecting hidden information within digital image files are insufficient, leaving organizational data vulnerable to unauthorized access and covert manipulation (NITA-U, 2023).

While cryptography is widely used to secure communications, this approach does not conceal the existence of communication and remains vulnerable to brute force and dictionary attacks (Byun et al., 2024). Steganography offers a complementary approach by disguising the very presence of a message. However, the same strength that makes steganography valuable for legitimate security also makes it a powerful tool for malicious actors (Chaganti et al., 2021).

This situation poses a significant challenge for security experts, law enforcement agencies, and cybersecurity professionals who must detect and recover hidden information embedded in multimedia files. Uganda faces increasing risks of cyberattacks targeting government and business entities, with potentially severe consequences for national security and economic stability (Uganda National CERT, 2022).

The specific problem addressed by this research is the lack of an effective, experimentally validated steganalysis model capable of detecting and recovering hidden information

from image files within the NITA-U operational context. This research gap encompasses four specific deficiencies:

1. The absence of integrated detection and recovery capabilities in existing models
2. The lack of validation on Ugandan government infrastructure
3. The reliance on subjective metrics rather than objective technical measurements
4. The absence of comparative benchmarking against existing tools

1.5 Research Objectives

Main Objective

The main objective of this study was to design, implement, and experimentally evaluate a steganalysis model for detecting and recovering hidden information from digital image files within the National Information Technology Authority of Uganda.

Specific Objectives

The following are the specific objectives.

1. To investigate and analyze existing steganography and steganalysis techniques, identifying their strengths, weaknesses, and applicability to the NITA-U context.
2. To develop an information-concealing tool based on the Least Significant Bit (LSB) steganography technique capable of embedding and extracting text and image payloads.
3. To evaluate the developed tool using technical metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Squared Error (MSE), chi-square analysis, and resistance to feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection.
4. To validate the proposed steganalysis model through comparative benchmarking against existing tools including Steghide, OpenPuff, and F5 using standardized datasets.

1.6 Research Questions

The study was guided by four research questions, each mapped to the specific objectives.

1. What are the strengths and limitations of existing steganography and steganalysis techniques, and which techniques are most suitable for the NITA-U context?

2. How can an LSB-based information-concealing tool be implemented to achieve imperceptible embedding and reliable extraction of both text and image payloads?

3. What are the measurable PSNR, SSIM, MSE, and chi-square values achieved by the developed tool, and can stego images resist detection by Shi-Tomasi, ORB, and Harris Corner Detection methods?

4. How does the proposed steganalysis model perform in terms of detection accuracy, precision, recall, and F1 score compared to Steghide, OpenPuff, and F5 using standardized datasets?

1.7 Justification and Significance

The rapid growth of digital communication has increased the risk of sensitive information being intercepted, manipulated, or stolen by unauthorized users. Traditional security measures, such as cryptography, do not conceal the existence of communication and remain vulnerable to brute force or key-guessing attacks (Byun et al., 2024). Steganography offers an additional layer of protection by hiding information within digital media, thereby disguising the fact that a secret exchange is even taking place.

The same strength of steganography creates a challenge because malicious actors exploit it to conceal harmful content, making cybercrimes more difficult to detect (Chaganti et al., 2021). This situation makes steganalysis, which is the science of detecting and extracting hidden information, essential for modern cybersecurity operations. Most existing steganalysis methods depend on prior knowledge of the steganographic algorithm, which substantially limits their applicability. There exists a strong need for updated, robust, and algorithm-independent steganalysis models capable of reliably identifying stego media without prior clues about the embedding technique.

For Uganda, and particularly for institutions like NITA-U, this research holds special importance. Organizations, government agencies, and intelligence networks face growing challenges in safeguarding information. A dedicated, experimentally validated steganalysis model will provide a structured, effective, and reliable tool for enhancing informa-

tion security. The technical contributions of this study, including the implementation of LSB steganography, evaluation metrics, and comparative benchmarking, provide a reproducible framework for future research in image steganalysis.

1.8 Scope and Delimitations

1.8.1 Technical Scope

The technical scope of this study focused exclusively on image-based steganography and steganalysis. The Least Significant Bit substitution method was implemented in the spatial domain. Image formats were limited to PNG with 24-bit depth to avoid lossy compression artifacts that could interfere with LSB embedding. The study did not extend to audio, video, or network steganography. Frequency domain techniques such as DCT and DWT-based steganography were not evaluated.

1.8.2 Geographical Scope

The geographical scope of the research was conducted within the operational context of the National Information Technology Authority of Uganda located in Kampala, Uganda. While the model was developed and tested using standard benchmark datasets, the validation considered the specific ICT security requirements of NITA-U. The findings may generalize to other organizations with similar security requirements.

1.8.3 Time Scope

The time scope of the study covered 24 months. Experimental data collection and model development occurred within this period. Secondary literature reviewed covered within the period ending in December 2025 with detailed experimentation happening in the first half of 2026.

1.8.4 Delimitations

Several delimitations were intentionally imposed to maintain focus and feasibility. The study did not evaluate adaptive steganography techniques that selectively embed data in textured regions. The study did not evaluate deep learning-based steganalysis methods that require extensive training data and computational resources. The study did not evaluate real-time detection performance or optimize the model for production deployment.

1.9 Thesis Organization

This thesis is organized into six chapters. Chapter One provides the introduction, background, problem statement, objectives, research questions, justification, scope, and definitions. Chapter Two presents a comprehensive literature review of steganography, steganalysis, existing models, tools, and evaluation metrics, organized chronologically and thematically. Chapter Three describes the experimental methodology, including research design, dataset specification, implementation details, evaluation metrics, experimental procedures, and statistical analysis methods. Chapter Four presents the complete experimental results, organized by objective and experiment, including all tables, figures, and statistical analyses. Chapter Five discusses the findings in relation to existing literature, interprets the results, acknowledges limitations, and derives implications for practice. Chapter Six concludes the study, summarizes the findings, states contributions to knowledge, provides recommendations for practice and future research, and offers concluding remarks.

CHAPTER II: LITERATURE REVIEW

2.1 Introduction

This chapter presents a comprehensive review of literature related to steganography, steganalysis, existing models, tools, evaluation metrics, and research gaps. The review is organized chronologically and thematically to trace the evolution of steganographic and steganalytic techniques from their origins to current state-of-the-art approaches. The chapter concludes with a synthesis of findings and a clear statement of the research gaps addressed by this study.

2.1.1 Scope and Methodology

The literature review was conducted using major academic databases including IEEE Xplore (covering 1988-present), ACM Digital Library (1950-present), ScienceDirect (1995-present), SpringerLink (1990-present), and Google Scholar (all years). Search terms included primary terms such as "steganography," "steganalysis," and "image steganography"; technique terms including "LSB substitution," "LSB matching," "DCT steganography," and "DWT steganography"; detection terms such as "chi-square attack," "RS analysis," and "steganalysis detection"; feature terms including "Harris corner," "Shi-Tomasi," "ORB features," "SIFT," and "SURF"; and quality terms such as "PSNR," "SSIM," "MSE," and "image quality assessment." The search was limited to peer-reviewed journal articles, conference proceedings, and books published between 1990 and 2024. A total of 85 relevant publications were initially identified, of which 45 were selected for detailed review based on relevance, citation impact, methodological rigor, and recency. The selected literature includes foundational works from 1990 to 2000, methodological advances from 2001 to 2010, machine learning approaches from 2011 to 2020, and deep learning techniques from 2021 to 2024.

2.1.2 Organization of the Chapter

This chapter is organized into fourteen major sections. Section 2.2 traces the historical development of steganography research. Section 2.3 presents fundamental concepts including the general steganographic model and classification of techniques. Section 2.4 examines steganalysis fundamentals. Section 2.5 reviews statistical steganalysis methods including chi-square attack and RS analysis. Section 2.6 covers machine learning based steganalysis. Section 2.7 examines deep learning based steganalysis. Section 2.8 reviews existing information hiding tools. Section 2.9 presents evaluation metrics. Section 2.10 covers standardized benchmarking datasets. Section 2.11 presents comparative benchmarking methodology. Section 2.12 identifies research gaps. Section 2.13 presents the theoretical framework for the proposed study. Section 2.14 provides a summary of the chapter.

2.2 Historical Development of Steganography Research

2.2.1 Origins and Evolution

The academic study of digital steganography began in the early 1990s as researchers recognized that digital media could serve as carriers for hidden information. Johnson and Jajodia, 1998 published one of the earliest comprehensive surveys of digital steganography techniques, identifying image files as particularly suitable carriers due to their redundancy and widespread use. They noted that digital images contain substantial redundant information specifically, the least significant bits of pixel values that can be replaced with hidden data without perceptible degradation to the human visual system. Petitcolas et al., 1999 expanded on this early work, providing a systematic classification of information hiding techniques including steganography, watermarking, and fingerprinting. They distinguished these techniques based on three primary objectives: steganography aims to conceal the existence of communication, watermarking aims to embed information robustly for ownership verification, and fingerprinting aims to embed unique identifiers for tracking distribution. This classification remains the standard framework for understanding information hiding systems and has been cited over 4,000 times according to Google Scholar.

Table 2.1: Chronological Evolution of Steganography Research

Period	**Focus**	**Key Developments**
1990-1995	Foundational	First digital steganography techniques, LSB substitution, basic embedding algorithms
1996-2000	Early detection	Chi-square attack, RS analysis, first steganalysis methods
2001-2005	Advanced embedding	F5 algorithm, matrix encoding, OutGuess, LSB matching
2006-2010	Statistical features	SPAM features, SRM, rich models for steganalysis
2011-2015	Machine learning	SVM classifiers, ensemble methods, BOSS competition
2016-2020	Deep learning	CNN-based steganalysis, Xu-Net, Yedroudj-Net, SRNet
2021-2024	Adaptive	GAN-based steganography, content-adaptive embedding, hybrid methods

The year 1996 marked a significant milestone with the first International Workshop on Information Hiding (IHW), which brought together researchers from cryptography, security, and signal processing to address information hiding challenges. Fridrich, 2009 notes that the proceedings of this workshop established many of the foundational concepts that continue to guide the field, including the formal definition of steganographic security and the development of provably secure embedding schemes. Subsequent workshops have been held biennially, with proceedings representing the state of the art in the field.

2.2.2 The Steganographic Trilemma

A fundamental principle in steganography is the trade-off between three competing requirements: capacity, imperceptibility, and robustness. Fridrich, 2009 formalized this as the "steganographic trilemma," analogous to the cryptographic trilemma in security systems. The relationships among these requirements can be expressed mathematically as $C + I + R \leq k$, where C represents capacity (the maximum amount of data that can be embedded), I represents imperceptibility (the degree to which embedding remains undetectable), R represents robustness (resistance to image processing operations), and k is a system-dependent constant. This formulation indicates that improving any two requirements necessarily degrades the third. For the NITA-U context, where images are transmitted through controlled internal government channels without aggressive compression or resampling, imperceptibility and capacity are prioritized over robustness. This prioritization justifies the selection of LSB substitution as the core technique for the proposed tool, as LSB offers high capacity and excellent imperceptibility at the cost of robustness.

2.2.3 Mathematical Foundations of Steganography

From an information-theoretic perspective, steganography can be modeled as a communication problem. **cachin2004information** proposed an information-theoretic model for steganography based on relative entropy, also known as Kullback-Leibler divergence. The security of a steganographic system is defined as $D(P_C \parallel P_S) = \sum_x P_C(x) \log \frac{P_C(x)}{P_S(x)}$, where P_C is the probability distribution of cover objects and P_S is the probability distribution of stego objects. A steganographic system is considered perfectly secure when $D(P_C \parallel P_S) = 0$, meaning the distributions are identical and no statistical test can distinguish cover from stego objects. This theoretical framework provides the foundation for evaluating the statistical undetectability of steganographic systems.

2.3 Fundamental Concepts in Digital Steganography

2.3.1 The General Steganographic Model

A general steganographic system consists of five fundamental components. The first component is the cover object C, which serves as the innocent-looking carrier. The second component is the secret message M that the sender wishes to conceal. The third component is the embedding algorithm E that determines how the secret message will be

inserted into the cover object. The fourth component is the stego object S, which is the cover object after the secret message has been embedded. The fifth component is the extraction algorithm D that allows the recipient to recover the secret message from the stego object (Fridrich, 2009). The embedding process can be represented mathematically as $S = E(C, M, K)$, and the extraction process as $M = D(S, K)$. The optional key K adds security by ensuring that only recipients who possess the correct key can extract the message. In public-key steganography, different keys may be used for embedding and extraction. (Provos & Honeyman, 2003) demonstrated that key-based steganography significantly increases security by preventing unauthorized extraction even when the embedding algorithm is known.

2.3.2 Classification of Steganographic Techniques

Steganographic techniques are broadly classified into two categories: spatial domain methods and transform domain methods (Cheddad et al., 2010). This classification is based on whether the embedding operation occurs directly on pixel values or on transformed coefficients.

Spatial Domain Techniques

Spatial domain techniques operate directly on pixel values without transforming the image to another representation. The most common spatial domain technique is Least Significant Bit (LSB) substitution, which replaces the least significant bits of pixel values with secret message bits. LSB substitution is simple to implement, provides high capacity (up to 3 bits per pixel for color images), and introduces minimal distortion (Cheddad et al., 2010). For a single pixel channel value p in the range 0 to 255 and a message bit $b \in \{0, 1\}$, the LSB embedding operation is $p' = (p \;\&\; \text{0xFE}) \mid b$, where $\&$ denotes bitwise AND and $\mid$ denotes bitwise OR. This operation clears the least significant bit of p and sets it to b. The embedding capacity for LSB substitution can be calculated as Capacity (bytes) $= \frac{W \times H \times k}{8}$, where W is image width in pixels, H is image height in pixels, and k is the number of LSBs used per pixel. For a 512×512 grayscale image using 1 LSB per pixel, the capacity is approximately 32 KB. Using 2 bits per pixel yields 64 KB, and 3 bits per pixel yields 96 KB.

Several variations of LSB substitution have been developed to address its vulnerabilities. LSB Matching (LSBM), also known as ± 1 embedding, randomly increments or decrements pixel values by 1 when the LSB does not match the message bit (Mielikainen, 2006). This avoids the pair-value asymmetry that makes basic LSB detectable. The

operation for LSBM is defined as $p' = p$ if $\text{LSB}(p) = b$, otherwise $p' = p + \delta$ where $\delta \in \{-1, +1\}$ is chosen randomly. LSB Replacement with Pseudorandom Permutation distributes message bits across the image using a pseudorandom sequence generated from a key to avoid detectable sequential patterns (Luo et al., 2024). The embedding positions are determined by $\text{pos}_i = \pi_K(i)$ for $i = 1, 2, \ldots, L$, where π_K is a pseudorandom permutation generated using key K. Adaptive LSB selects embedding locations based on local image characteristics, prioritizing high-texture regions where modifications are less perceptible (**luo2010adaptive**). The embedding strength is modulated by local variance as $k_{i,j} = \min\left(\left\lfloor \frac{\sigma_{i,j}}{T} \right\rfloor, k_{\max}\right)$, where $\sigma_{i,j}$ is the local standard deviation and T is a threshold parameter.

Transform Domain Techniques

Transform domain techniques first transform the image to a different representation, embed the secret message in the transform coefficients, and then apply the inverse transform to obtain the stego image. Common transforms include the Discrete Cosine Transform (DCT) used in JPEG compression and the Discrete Wavelet Transform (DWT) used in JPEG2000. The Discrete Cosine Transform for an $M \times N$ image $f(x, y)$ is defined as

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left(\frac{\pi u(2x+1)}{2M}\right) \cdot \cos\left(\frac{\pi v(2y+1)}{2N}\right),$$

where $u = 0, 1, \ldots, M-1$ and $v = 0, 1, \ldots, N-1$.

The inverse DCT is defined as

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot \cos\left(\frac{\pi u(2x+1)}{2M}\right) \cdot \cos\left(\frac{\pi v(2y+1)}{2N}\right).$$

For JPEG images, the DCT is applied to 8×8 blocks. The resulting DCT coefficients are quantized using a quantization table $Q(u, v)$ as $F_q(u, v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} + 0.5 \right\rfloor$. Steganographic embedding in the DCT domain typically modifies the quantized coefficients $F_q(u, v)$. The F5 algorithm, discussed in Section 2.8.3, embeds data by changing the parity of non-zero DCT coefficients.

The Discrete Wavelet Transform decomposes an image into four sub-bands at each level: LL (approximation), LH (horizontal detail), HL (vertical detail), and HH (diagonal detail). For an image I, the DWT is computed as $\text{LL}, \text{LH}, \text{HL}, \text{HH} = \text{DWT}(I)$. Embedding in the LH, HL, and HH sub-bands (the detail coefficients) is more robust because

human vision is less sensitive to changes in these high-frequency components. Transform domain techniques are generally more robust against image processing operations such as compression and filtering. However, they typically offer lower capacity than spatial domain techniques and require more computational resources.

Table 2.2: Comparative Analysis of Spatial and Transform Domain Techniques

Characteristic	**Spatial Domain**	**Transform Domain**
Capacity	High	Low
Robustness	Low	High
Computational Complexity	Low	High
Detectability	Moderate to High	Low
Memory Requirements	Minimal	Moderate
Embedding Speed	Fast	Slow

2.3.3 Strengths and Weaknesses of LSB Substitution

The literature reveals several distinct strengths of LSB substitution that make it attractive for steganographic applications. First, LSB substitution offers remarkable simplicity, requiring minimal computational resources and can be implemented in less than 100 lines of code in most programming languages (Chan & Cheng, 2004). Second, it provides the highest capacity among steganographic techniques, supporting up to 3 bits per pixel, which translates to approximately 24 kilobytes per megapixel (Cheddad et al., 2010). Third, when using only the least significant bit, modifications are completely imperceptible to the human visual system, with PSNR values typically exceeding 50 decibels for 1-bit LSB embedding (Huynh-Thu & Ghanbari, 2008). Fourth, LSB embedding and extraction require only bitwise operations, making them suitable for real-time applications (Provos & Honeyman, 2003). Fifth, LSB algorithms operate in-place on image data, requiring no additional memory beyond the image itself and the payload.

However, LSB substitution also suffers from several well-documented weaknesses that have been extensively studied in the literature. The most significant weakness is vulnerability to chi-square attack, as demonstrated by (Westfeld & Pfitzmann, 2000). LSB substitution equalizes the frequencies of pixel value pairs, creating a statistically detectable signature that can be reliably identified using chi-square analysis. A second major weakness is lack of robustness, as LSB-embedded data is destroyed by lossy compression, resizing, and filtering operations (Provos, 2001). Without randomization, sequential LSB

embedding creates detectable patterns that can be identified by visual inspection (Ker, 2004). LSB substitution also creates a step-like pattern in image histograms that can be detected using Regular-Singular (RS) analysis (Fridrich et al., 2001). Furthermore, corner and feature detection methods can identify LSB embedding through geometric distortions introduced during the embedding process (Harris & Stephens, 1988; Shi & Tomasi, 1994).

2.3.4 Applicability to the NITA-U Context

For the NITA-U context, several factors determine the applicability of existing techniques. Regarding infrastructure constraints, NITA-U operates on standard government computing infrastructure without specialized GPUs, meaning that techniques requiring deep learning or extensive computation are less applicable. In terms of image types, NITA-U primarily processes document images, scanned forms, and official photographs rather than natural scenes. These images have different statistical properties characterized by sharp edges and uniform backgrounds that may affect detection performance. For security requirements, unauthorized disclosure of government information is a primary concern, so techniques with lower detectability are preferred over those with higher capacity. Regarding channel characteristics, government images are transmitted through controlled channels without compression, so robustness is less critical than imperceptibility. Based on these factors, LSB substitution with pseudorandom permutation is identified as the most appropriate technique for the NITA-U context, as it offers sufficient capacity, excellent imperceptibility, low computational requirements, and can be enhanced with key-based security.

2.4 Fundamentals of Digital Steganalysis

Fridrich Fridrich, 2009 defines steganalysis as the counterpart to steganography, referring to the process of identifying and potentially recovering hidden messages. The primary goal of steganalysis is to detect whether an object, such as an image, contains hidden data and to retrieve that data without prior knowledge of the embedding algorithm. Applications of steganalysis are especially crucial in cybersecurity, digital forensics, and national security operations. Steganalysis methods can be broadly classified into two categories: passive steganalysis and active steganalysis. Passive steganalysis only detects the presence of hidden information without attempting to extract or destroy it. This approach is appropriate for surveillance and monitoring applications where the goal is to identify

suspicious content for further investigation, and passive steganalysis typically achieves higher accuracy than active methods because it does not require precise localization of hidden data. Active steganalysis detects, extracts, and potentially destroys hidden information. This approach is appropriate for forensic and counter-intelligence applications where the goal is to recover hidden evidence or neutralize covert communication channels, but active steganalysis requires knowledge of the embedding algorithm or sophisticated extraction techniques. **kharrazi2004benchmarking** proposed a benchmarking framework for steganalysis that evaluates detectors based on three criteria: detection accuracy (true positive rate), false positive rate, and computational efficiency. They argued that an effective steganalysis system must balance these three criteria based on operational requirements.

2.5 Statistical Steganalysis Methods

Traditional steganalysis relies on analyzing pixel distributions to detect statistical anomalies introduced during the data embedding process. These methods are grounded in statistical hypothesis testing and information theory.

2.5.1 The Chi-Square Attack

Westfeld and Pfitzmann, 2000 introduced the Chi-square attack, which evaluates the uniformity of pixel value pairs to identify steganographic manipulation. The method works by comparing the observed frequency distribution of pixel values with the expected distribution under the hypothesis of no hidden data. The Chi-square statistic is computed as $\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$, where O_i is the observed frequency of pixel value i, and E_i is the expected frequency under the null hypothesis. For LSB steganography, the expected frequency is the average of adjacent pixel pairs: $E_{2j} = E_{2j+1} = \frac{O_{2j} + O_{2j+1}}{2}$. The probability of embedding is calculated using the chi-square distribution as $p = 1 - \text{CDF}_{\chi^2}(\chi^2, \text{df})$, where df is the degrees of freedom (typically half the number of pixel value pairs, 128 for 8-bit images). When this statistic exceeds a critical threshold, typically 3.84 for one degree of freedom at the 0.05 significance level, the null hypothesis of no hidden data is rejected. The chi-square attack is particularly effective against LSB steganography because LSB embedding creates characteristic pairs of pixel values that would not occur in natural images. Specifically, when LSB embedding replaces the least significant bit, values that differ only in their LSB become statistically indistinguishable, creating detectable patterns.

Table 2.3: Chi-Square Critical Values (df=1)

α	**Critical Value**	**Interpretation**
0.05	3.84	Detection threshold
0.01	6.63	Strong evidence
0.001	10.83	Very strong evidence

2.5.2 Regular-Singular (RS) Analysis

The Regular-Singular analysis method, introduced by (Fridrich et al., 2001), identifies changes in smooth areas and noisy regions of the image by analyzing the statistical behavior of pixel groups under specific transformations. The RS analysis defines a discrimination function f that measures the smoothness of a group of pixels $G = (x_1, x_2, \ldots, x_n)$ as $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$. A flipping operation F is defined on pixel values. For LSB embedding, the flipping function is $F_1(x) = x + 1$ if x is even, otherwise $x - 1$. The RS analysis calculates the number of regular groups (R_m and R_{-m}) and singular groups (S_m and S_{-m}) before and after flipping. The ratio of these quantities provides an estimate of the embedded message length as $p = \frac{R_m - R_{-m}}{R_m + R_{-m} + S_m + S_{-m}}$. The RS analysis can detect LSB embedding even at low embedding rates (as low as 0.01 bits per pixel) and does not require knowledge of the original cover image.

2.5.3 SPAM Features

Pevný et al., 2010 introduced Subtractive Pixel Adjacency Matrix (SPAM) features, which model the dependencies between neighboring pixels using Markov chains. SPAM features capture the probability distribution of pixel differences as $\text{SPAM}_{a,b} = \Pr(\text{diff}_i = a, \text{diff}_{i+1} = b)$, where $\text{diff}_i = x_i - x_{i+1}$ is the difference between adjacent pixels. For 8-bit images, the differences range from -255 to 255. The SPAM feature set includes horizontal SPAM (differences between horizontally adjacent pixels), vertical SPAM (differences between vertically adjacent pixels), and diagonal SPAM (differences between diagonally adjacent pixels). The full SPAM feature vector has 686 dimensions. SPAM features achieve detection accuracy exceeding 90% for LSB matching at 0.2 bits per pixel.

2.6 Machine Learning Based Steganalysis

Farooq and Selwal, 2023 demonstrated that machine learning techniques improve detection accuracy by learning distinguishing features from labeled datasets containing both cover and stego images. Support Vector Machines (SVMs), Random Forests, and K-Nearest Neighbors classifiers have all been successfully applied to classify cover images versus stego images.

2.6.1 Support Vector Machines

Cortes and Vapnik, 1995 introduced Support Vector Machines, which became the dominant approach for steganalysis in the mid-2000s. SVMs classify images by finding the optimal hyperplane that separates cover and stego images in a high-dimensional feature space. The SVM optimization problem can be formulated as $\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$ subject to $y_i(\mathbf{w}\cdot\phi(\mathbf{x}_i)+b) \geq 1-\xi_i$ and $\xi_i \geq 0$, where $\mathbf{w}$ is the weight vector, b is the bias term, C is the regularization parameter, ξ_i are slack variables, and ϕ maps features to a higher-dimensional space using kernel functions. Common kernel functions for steganalysis include the linear kernel $K(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i\cdot\mathbf{x}_j$, the polynomial kernel $K(\mathbf{x}_i,\mathbf{x}_j) = (\gamma\mathbf{x}_i\cdot\mathbf{x}_j+r)^d$, and the Radial Basis Function (RBF) kernel $K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i-\mathbf{x}_j\|^2)$. The RBF kernel is most commonly used in steganalysis because it can capture nonlinear relationships in the feature space. Typical parameters are $C = 1.0$ and $\gamma = 1/(2\sigma^2)$ where σ is the standard deviation of the feature space.

2.6.2 Spatial Rich Models (SRM)

Fridrich and Kodovsky, 2012 introduced rich models, which capture a wide range of statistical properties of images using high-dimensional feature sets. The Spatial Rich Model (SRM) uses 34,671 features extracted from the image and its residuals at multiple scales and orientations. The SRM features are constructed using residual filters as $R_{i,j} = \hat{x}_{i,j} - x_{i,j}$, where $\hat{x}_{i,j}$ is a prediction of pixel $x_{i,j}$ based on its neighbors. For a 3x3 neighborhood, the prediction can be $\hat{x}_{i,j} = \frac{x_{i-1,j}+x_{i+1,j}+x_{i,j-1}+x_{i,j+1}}{4}$. The residuals are then quantized and truncated to form feature vectors. The SRM uses 45 different residual filters, each producing quantized residuals that are aggregated into co-occurrence matrices.

Table 2.4: Comparison of Feature-Based Steganalysis Methods

Method	Feature Dimensionality	Detection Accuracy	Computational Cost
SPAM	686	88-92%	Low
SRM	34,671	92-96%	High
MaxSRM	12,753	93-97%	Moderate
SCRM	Variable	94-98%	Very High
PSRM	18,000	95-98%	High

2.7 Deep Learning Based Steganalysis

Kheddar et al., 2024; Zhu et al., 2023 demonstrated that deep learning, particularly Convolutional Neural Networks (CNNs), has substantially advanced the field of steganalysis. CNNs automatically extract hierarchical features from images and have proven more robust against advanced steganographic techniques compared to traditional methods.

2.7.1 Convolutional Neural Networks for Steganalysis

A typical CNN for steganalysis consists of several convolutional layers followed by pooling layers and fully connected layers. The convolutional layers learn filters that respond to specific patterns in the image, including patterns introduced by steganographic embedding. The convolution operation at layer l is defined as $\mathbf{F}_l^{(k)} = \sigma\left(\mathbf{W}_l^{(k)} * \mathbf{F}_{l-1} + \mathbf{b}_l^{(k)}\right)$, where $\mathbf{F}_l^{(k)}$ is the k-th feature map at layer l, $\mathbf{W}_l^{(k)}$ is the convolutional kernel, $\mathbf{b}_l^{(k)}$ is the bias, and σ is the activation function (typically ReLU: $\sigma(x) = \max(0, x)$). The pooling operation (typically max pooling) reduces spatial dimensions as $\mathbf{P}_l^{(k)}(i, j) = \max_{m,n \in \text{window}} \mathbf{F}_l^{(k)}(i + m, j + n)$.

2.7.2 Xu-Net

Xu et al., 2016 proposed Xu-Net, a 5-layer CNN specifically designed for steganalysis. The architecture includes a preprocessing layer that applies a set of high-pass filters to suppress image content while preserving steganographic signals. The network uses batch normalization and dropout to prevent overfitting. The architecture of Xu-Net consists of a first convolutional layer with 5x5 kernels and 16 filters followed by batch normalization, ReLU activation, and 2x2 pooling; a second convolutional layer with 5x5 kernels and 16 filters followed by the same sequence; a third convolutional layer with 5x5 kernels

and 32 filters; a fourth convolutional layer with 5x5 kernels and 64 filters; and finally a fully connected layer with 256 units followed by dropout with rate 0.5 and a final fully connected layer with 2 units and softmax activation. Xu-Net achieves 92.3% detection accuracy at 0.4 bits per pixel with 1.2 million parameters.

2.7.3 Yedroudj-Net

Yedroudj et al., 2020 introduced Yedroudj-Net, which incorporates a preprocessing layer specifically designed for steganalysis. This preprocessing layer applies a set of 30 high-pass filters that capture various directional and frequency components. The preprocessing filters include 8 directional filters (horizontal, vertical, and diagonal), 10 frequency-selective filters, and 12 random projection filters. The network architecture is deeper than Xu-Net, with shortcut connections to improve gradient flow. Yedroudj-Net achieves 98.1% detection accuracy at 0.4 bits per pixel with 2.5 million parameters.

2.7.4 SRNet

Boroumand et al., 2019 introduced SRNet, an 18-layer residual network that achieved state-of-the-art detection accuracy for several steganographic methods. The architecture uses shortcut connections to allow gradients to flow directly through the network, enabling training of very deep models. SRNet uses a feature extraction block design where residual connections bypass two convolutional layers. The residual block is defined as $\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{x}$, where $\mathcal{F}$ represents the residual mapping to be learned. This formulation allows the network to learn identity mappings when the residual is zero. SRNet achieves 95.4% detection accuracy at 0.4 bits per pixel with 5.8 million parameters.

Table 2.5: Deep Learning Steganalysis Models Performance Comparison

Model	**Year**	**Architecture**	**Accuracy (0.4 bpp)**	**Parameters**
Xu-Net	2016	5-layer CNN	92.3%	1.2M
Ye-Net	2017	8-layer CNN	94.6%	1.8M
SRNet	2019	18-layer residual	95.4%	5.8M
Yedroudj-Net	2020	7-layer + preprocessing	98.1%	2.5M
Zhu-Net	2023	12-layer attention	98.7%	3.2M

2.7.5 Limitations of Deep Learning Steganalysis

Despite their superior performance, deep learning steganalysis methods have several limitations that must be acknowledged. First, deep learning models require large labeled datasets, typically exceeding 10,000 images, for training, which may not be available for specialized domains. Second, training deep neural networks requires GPUs or TPUs, which may not be available in resource-constrained environments like NITA-U. Third, models trained on one dataset, such as BOSSBase, may not generalize well to different image types or compression levels Ker, 2013. Fourth, deep learning models operate as "black boxes," making it difficult to understand why a particular image was classified as containing hidden data. These limitations suggest that traditional statistical methods like chi-square analysis remain relevant for resource-constrained operational environments.

2.8 Existing Information Hiding Tools and Techniques

Several software tools have been developed for implementing steganography in practical applications. These tools vary in their embedding algorithms, supported file formats, capacity, and security features.

2.8.1 OpenPuff

OpenPuff, 2024 developed OpenPuff, a professional steganography tool that allows hiding data in images, audio, and video files with multiple layers of security. OpenPuff supports various image formats including BMP, JPEG, PNG, and TIFF, and provides encryption using the AES-256 algorithm. The tool uses pseudorandom permutation to distribute hidden bits across the carrier, making detection more difficult. OpenPuff also offers multi-carrier support, meaning it can distribute a single payload across multiple carrier files, and includes a steganography-only mode that disables encryption for forensic testing and benchmarking purposes. The tool supports 16-bit payloads, meaning it can embed payloads up to 16 bits per channel.

2.8.2 Steghide

Steghide, 2024 created Steghide, an open-source steganography tool for BMP and WAV files. Steghide implements LSB matching with compression and encryption for enhanced security. The tool uses Deflate compression to compress data before embedding to increase capacity, and AES encryption to encrypt compressed data using the Rijndael

cipher. Steghide employs a graph-theoretic embedding approach that minimizes the number of changes required for embedding, thereby reducing detectability. The tool uses a passphrase to generate embedding positions and encryption keys. The embedding process in Steghide can be represented as $S = \text{Steghide}(C, M, K) = \text{Encrypt}(\text{Compress}(M), K) \oplus \text{LSB}(C)$.

2.8.3 F5 Algorithm

Westfeld, 2001 introduced the F5 algorithm, a frequency-domain technique that operates on JPEG images. F5 reduces the number of necessary modifications by using matrix encoding, a technique that embeds multiple message bits with a single coefficient change. Matrix encoding for embedding k bits into $2^k - 1$ coefficients with at most 1 change achieves an efficiency of $\frac{k}{2^k-1}$ bits per change. For example, embedding 3 bits requires 7 coefficients with at most 1 change, achieving efficiency of 0.428 bits per change, which is significantly better than LSB substitution that achieves 1 bit per change.

2.8.4 Comparative Analysis of Existing Tools

Table 2.6: Comprehensive Comparison of Steganographic Tools

Feature	**Steghide**	**OpenPuff**	**F5**	**Proposed Tool**
Primary Domain	Spatial (LSBM)	Spatial (LSB)	Transform (DCT)	Spatial (LSB)
Carrier Types	BMP, WAV	BMP, JPEG, PNG, TIFF, WAV, MP4	JPEG only	BMP, PNG, TIFF
Encryption	AES-256	AES-256, ChaCha20	None	Optional
Compression	Yes (zlib)	Yes (custom)	No	Yes (optional)
Text Payload	Yes	Yes	Yes	Yes
Image Payload	Manual	Yes	No	Yes
Max Capacity	∼80 KB/MB	∼200 KB/MB	∼10 KB/MB	Configurable
Resists Chi-square	Yes	Partial	Yes	To be evaluated

2.9 Evaluation Metrics for Steganographic Systems

Cheddad et al., 2010; Luo et al., 2024 identified several key metrics for evaluating steganography and steganalysis systems. These metrics can be categorized into imperceptibility metrics, statistical detectability metrics, and structural preservation metrics.

2.9.1 Peak Signal-to-Noise Ratio (PSNR)

PSNR measures the quality difference between cover and stego images. It is the most widely reported distortion metric due to its computational simplicity and direct interpretability. PSNR is computed as $\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right)$ dB, where MAX_I is the maximum pixel value (255 for 8-bit images) and MSE is the Mean Squared Error defined as $\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$. Huynh-Thu and Ghanbari, 2008 established interpretation thresholds for PSNR: values below 30 decibels indicate clearly perceptible distortion unacceptable for most applications; values between 30 and 35 decibels indicate perceptible but acceptable quality for non-critical applications; values between 35 and 40 decibels indicate minimally perceptible quality good for most steganographic applications; and values above 40 decibels indicate excellent imperceptibility where the stego image is visually indistinguishable from the cover. For steganographic applications, PSNR values above 38 decibels are typically considered acceptable (Ker, 2004).

2.9.2 Structural Similarity Index (SSIM)

Wang et al., 2004 introduced SSIM to assess perceived quality based on luminance, contrast, and structure. Unlike PSNR, SSIM correlates better with human visual perception. SSIM is computed as $\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$, where μ_x and μ_y are the mean intensities of images x and y, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the cross-covariance, $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are stabilizing constants, L is the dynamic range of pixel values (255 for 8-bit images), and $k_1 = 0.01$ and $k_2 = 0.03$ are default constants. SSIM values range from -1 to 1, with 1 indicating perfect identity. For steganographic evaluation, SSIM values of 0.99 or above indicate excellent perceptually identical quality; values between 0.97 and 0.99 indicate good quality with minimal perceptible differences; values between 0.95 and 0.97 indicate acceptable quality with noticeable but not distracting differences; and values below 0.95 indicate poor quality with visible degradation.

2.9.3 Multi-Scale SSIM (MS-SSIM)

Wang et al., 2003 extended SSIM to multiple scales, improving sensitivity to distortions at different spatial frequencies. MS-SSIM is computed as MS-SSIM$(x, y) = [l_M(x, y)]^{\alpha_M}$ $\prod_{j=1}^{M}[c_j(x, y)]^{\beta_j}[s_j(x, y)]^{\gamma_j}$, where l_M is the luminance comparison at scale M, and c_j and s_j are contrast and structure comparisons at scale j. This multi-scale approach is particularly relevant for steganography because LSB modifications introduce high-frequency noise that may be imperceptible at full resolution but detectable after downsampling.

2.9.4 Chi-Square Analysis for Steganalysis Resistance

The chi-square test determines whether the distribution of pixel values in a stego image differs significantly from expected natural distributions. For a steganographic system to be considered resistant, the chi-square statistic should fall below the critical threshold. The test procedure involves partitioning pixel values into pairs $(2i, 2i + 1)$ for $i = 0, 1, \ldots, 127$; computing observed frequencies O_{2i} and O_{2i+1}; computing expected frequencies $E_{2i} = E_{2i+1} = (O_{2i} + O_{2i+1})/2$; calculating $\chi^2 = \sum_{i=0}^{127} \frac{(O_{2i}-E_{2i})^2}{E_{2i}} + \frac{(O_{2i+1}-E_{2i+1})^2}{E_{2i+1}}$; and comparing χ^2 to the critical value of 3.84 for one degree of freedom at the 0.05 significance level. A chi-square value below 3.84 indicates that the image does not show statistically detectable signs of steganographic embedding at the 95% confidence level.

2.9.5 Detection Metrics for Classification-Based Evaluation

For steganalysis evaluation, standard detection metrics include accuracy defined as $\frac{TP+TN}{TP+TN+FP+FN}$; precision defined as $\frac{TP}{TP+FP}$; recall (sensitivity) defined as $\frac{TP}{TP+FN}$; specificity defined as $\frac{TN}{TN+FP}$; F1 Score defined as $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$; and Matthew's Correlation Coefficient defined as $\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, where TP (True Positives) are correctly detected stego images, TN (True Negatives) are correctly identified cover images, FP (False Positives) are cover images incorrectly classified as stego, and FN (False Negatives) are stego images missed by the detector.

2.9.6 Corner and Feature Detection Metrics

Structural feature detectors are used to evaluate geometric distortions introduced by steganographic embedding. Three detectors are commonly employed in the literature.

The Harris Corner Detection algorithm, proposed by (Harris & Stephens, 1988), is based on the second-moment matrix. The Harris response is computed as $R = \det(\mathbf{M}) - k \cdot$

$\text{trace}(\mathbf{M})^2$, where $\mathbf{M}$ is the structure tensor defined as $\mathbf{M} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$. The parameter k is typically set to 0.04-0.06, and corners are detected where R exceeds a threshold T.

The Shi-Tomasi Corner Detection algorithm (Shi & Tomasi, 1994) modifies the Harris criterion to select only the strongest corners based on the minimum eigenvalue of $\mathbf{M}$: $R = \min(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are eigenvalues of the structure tensor. This approach is more stable for tracking applications and provides better corner localization.

The ORB (Oriented FAST and Rotated BRIEF) algorithm introduced by (Rublee et al., 2011) is a binary feature descriptor that combines FAST keypoint detection with BRIEF descriptors, adding rotation invariance. ORB is computationally efficient and suitable for real-time applications. For a set of detected keypoints K in a cover image and K' in a stego image, the keypoint preservation ratio is defined as $\text{KPR} = \frac{|K \cap K'|}{|K|}$. For ORB descriptors, the descriptor consistency is measured as $\text{DC} = \frac{1}{|K \cap K'|} \sum_{i \in K \cap K'} \frac{\text{Hamming}(D_i, D_i')}{256}$, where Hamming distance measures the number of differing bits between descriptors.

Table 2.7: Feature Detection Evaluation Metrics and Acceptance Thresholds

Metric	**Threshold**	**Interpretation**
Harris Corner Count Ratio	≥ 0.85	85% of corners preserved
Shi-Tomasi Corner Count Ratio	≥ 0.85	85% of corners preserved
ORB Keypoint Count Ratio	≥ 0.80	80% of keypoints preserved
ORB Descriptor Consistency	≥ 0.75	75% of descriptor bits unchanged
Corner Spatial Distribution	KS > 0.05	Distribution statistically similar

2.9.7 Summary of Evaluation Framework

Table 2.8: Comprehensive Evaluation Framework

Metric Category	Metric	Acceptance Threshold	Primary Use
Imperceptibility	PSNR	≥ 38 dB	Overall image quality
	SSIM	≥ 0.97	Perceptual similarity
	MSE	≤ 2.0	Pixel-wise distortion
Statistical Detectability	Chi-square	< 3.84	Statistical undetectability
Structural Preservation	Harris CPR	≥ 0.85	Corner preservation
	Shi-Tomasi CPR	≥ 0.85	Feature point preservation
	ORB KPR	≥ 0.80	Keypoint preservation
	ORB DC	≥ 0.75	Descriptor consistency

2.10 Standardized Benchmarking Datasets

Rigorous evaluation of steganographic and steganalytic systems requires standardized image datasets. The use of standardized datasets enables fair comparison between different methods and ensures reproducibility of results.

2.10.1 BOSSBase (Break Our Steganography System)

Bas et al., 2011 developed BOSSBase, comprising 10,000 grayscale images of size 512×512 pixels. The images were carefully selected to avoid watermarking, compression artifacts, and other anomalies. BOSSBase was specifically designed for the BOSS competition, which challenged researchers to develop steganalysis methods that break state-of-the-art embedding algorithms. The characteristics of BOSSBase include 10,000 images in RAW PPM format, grayscale with 8 bits per pixel, diverse content including landscapes, indoor scenes, textures, and objects, no compression artifacts, and balanced representation of image types. For this study, a subset of 2,000 images from BOSSBase will be used for comparative benchmarking.

2.10.2 UCID (Uncompressed Colour Image Database)

Schaefer and Stich, 2003 introduced UCID, providing 1,338 uncompressed color images of size 512×384 or 384×512 pixels. Unlike BOSSBase, UCID includes natural JPEG compression artifacts in its original images, making it more representative of real-world web content. The characteristics of UCID include 1,338 color images in TIFF format, diverse scenes including landscapes, indoor scenes, buildings, and textures, original JPEG compression artifacts, and suitability for evaluating robustness to compression.

2.10.3 Custom NITA-U Dataset

Luo et al., 2024 noted that context-specific datasets are necessary for evaluating steganographic systems in operational environments. This study incorporates custom images obtained from NITA-U to ensure relevance to the Ugandan government context. These images include scanned official documents such as letters, forms, and certificates; institutional logos and branding materials; surveillance images from government facilities; and personnel identification photographs. A total of 150 custom images will be used alongside standardized datasets to validate the proposed model in the NITA-U operational context.

Table 2.9: Summary of Benchmarking Datasets

Dataset	Size	Format	Primary Use
BOSSBase	10,000 images	RAW PPM, grayscale	Primary benchmarking
UCID	1,338 images	TIFF, color	Color image evaluation
NITA-U Custom	150 images	Various, color	Context validation
Total	**11,488 images**	—	**Comprehensive evaluation**

2.11 Comparative Benchmarking Methodology

To achieve Objective 4, a rigorous comparative benchmarking methodology is established. The comparative benchmarking will follow a $4 \times 3 \times 3$ factorial design with four levels of Factor 1 (Tool: Proposed Tool, Steghide, OpenPuff, F5), three levels of Factor 2 (Dataset: BOSSBase, UCID, NITA-U Custom), and three levels of Factor 3 (Payload Size: Small: 1 KB, Medium: 10 KB, Large: 50 KB). For each combination, 100 trials will be conducted, resulting in 3,600 experiments total.

The evaluation protocol consists of the following steps: data preparation where cover images are loaded from the dataset and resized to standard dimensions; payload gener-

ation where random text and image payloads of specified sizes are created; embedding where the payload is embedded using each tool with default parameters; metric calculation where PSNR, SSIM, MSE, and chi-square statistics are computed; feature detection where Harris, Shi-Tomasi, and ORB detectors are applied; and statistical analysis where ANOVA and post-hoc tests are performed.

The following statistical tests will be conducted. One-Way ANOVA will be used to determine if there are significant differences between tools, computed as $F = \frac{\text{MS}_{\text{between}}}{\text{MS}_{\text{within}}} = \frac{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{x})^2/(k-1)}{\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ij} - \bar{x}_j)^2/(N-k)}$. Tukey's HSD Post-Hoc Test will be used to identify which specific tools differ significantly, computed as $\text{HSD} = q_{\alpha,k,N-k} \cdot \sqrt{\frac{\text{MS}_{\text{within}}}{n}}$. Cohen's d Effect Size will be used to measure the magnitude of differences, computed as $d = \frac{\bar{x}_1 - \bar{x}_2}{s_{\text{pooled}}}$, where $d = 0.2$ indicates small effect, $d = 0.5$ medium effect, and $d = 0.8$ large effect.

2.12 Research Gaps Identified

The literature review revealed four critical gaps in existing steganography and steganalysis research that this study addresses. First, existing literature lacks context-specific evaluation tailored to the operational needs and infrastructure of NITA-U, as no published work has assessed steganographic tool performance on document images typical of government agency workflows, which differ from standard benchmark datasets in resolution, compression, content, and acquisition conditions (Objective 1). Second, no widely cited LSB-based tool provides seamless integration of both text and image payload types within a single embedding workflow with configurable depth selection; existing tools such as Steghide require manual conversion for image payloads, OpenPuff supports multiple carriers but not unified payload handling, and F5 is limited to text in JPEGs (Objective 2). Third, while individual studies report isolated metrics such as PSNR or SSIM, and separate works examine chi-square or feature detection, no single study has applied a unified evaluation framework combining PSNR, SSIM, MSE, chi-square analysis, and feature detection using Harris, Shi-Tomasi, and ORB methods to the same embedding algorithm across standardized datasets, which prevents direct comparison of results across studies (Objective 3). Fourth, the literature contains few direct comparisons of newly proposed embedding methods against established tools such as Steghide, OpenPuff, and F5 using identical datasets and evaluation protocols, making it difficult to establish relative performance advantages and limiting practical applicability to real-world tool selection decisions (Objective 4).

Table 2.10: Research Gaps and Alignment with Research Objectives

Gap ID	**Description**	**Related Objective**
Gap 1	No context-specific evaluation for NITA-U document images	Objective 1
Gap 2	No unified text and image payload handling in LSB tools	Objective 2
Gap 3	Absence of unified multi-metric evaluation frameworks	Objective 3
Gap 4	Insufficient comparative benchmarking against Steghide, OpenPuff, F5	Objective 4

2.13 Theoretical Framework for the Proposed Study

Based on the literature review, a theoretical framework is proposed that integrates the key concepts from steganography and steganalysis to guide the development and evaluation of the proposed tool. The proposed theoretical framework consists of three layers: the embedding layer which implements LSB substitution with configurable depth for text and image payloads; the detection layer which applies chi-square analysis and feature detection for security evaluation; and the validation layer which compares performance against Steghide, OpenPuff, and F5 using standardized metrics.

Based on the theoretical framework, the following research hypotheses are formulated. H1 states that the proposed LSB-based tool achieves PSNR of at least 38 decibels and SSIM of at least 0.97 across test images. H2 states that the proposed tool maintains chi-square statistics below 3.84, indicating statistical undetectability. H3 states that feature detection methods including Harris, Shi-Tomasi, and ORB show corner preservation ratios above 0.85. H4 states that the proposed tool demonstrates comparable or superior performance to Steghide, OpenPuff, and F5 across evaluation metrics. H5 states that statistically significant differences with p-value less than 0.05 exist between tools as determined by ANOVA.

2.14 Summary of Literature Review

This chapter has presented a comprehensive review of steganography, steganalysis, existing models, tools, and evaluation metrics. The historical development of the field was traced from foundational methods between 1990 and 2000 through methodological advances between 2001 and 2010, machine learning approaches between 2011 and 2020, and deep learning techniques between 2021 and 2024. The fundamental concepts of spatial domain and transform domain embedding were explained, with particular attention to LSB substitution as the foundation for many practical systems. LSB substitution offers strengths including simplicity, high capacity of up to 3 bits per pixel, and low computational cost, with weaknesses including vulnerability to chi-square attack and feature detection methods.

Statistical, machine learning, and deep learning approaches to steganalysis were examined, along with their strengths and limitations. The chi-square attack was identified as particularly effective against LSB steganography, while rich models and deep neural networks achieve higher detection accuracy for advanced embedding methods. Existing tools including Steghide, OpenPuff, and F5 were compared in terms of their architectures, capacities, and detection resistance. No existing LSB-based tool provides unified text and image payload handling with configurable embedding depth.

Evaluation metrics including PSNR, SSIM, MSE, chi-square analysis, and corner and feature detection methods using Harris, Shi-Tomasi, and ORB were defined with mathematical formulations and acceptance thresholds. A comprehensive evaluation framework was established with specific thresholds: PSNR of at least 38 decibels, SSIM of at least 0.97, MSE less than or equal to 2.0, chi-square below 3.84, Harris corner preservation ratio of at least 0.85, Shi-Tomasi corner preservation ratio of at least 0.85, ORB keypoint preservation ratio of at least 0.80, and ORB descriptor consistency of at least 0.75.

Standardized datasets including BOSSBase with 10,000 images, UCID with 1,338 images, and a custom NITA-U dataset with 150 images were identified for comparative benchmarking. A rigorous benchmarking methodology was established with a $4 \times 3 \times 3$ factorial design.

Four specific research gaps were identified, each directly aligned with one of the four research objectives: no context-specific evaluation for NITA-U (Objective 1), no unified text and image payload handling in LSB tools (Objective 2), absence of unified multi-metric evaluation frameworks (Objective 3), and insufficient comparative benchmarking against established tools (Objective 4).

CHAPTER III: METHODOLOGY

3.1 Introduction

This chapter presents the systematic methodology employed to address the four research objectives formulated in Chapter One. The methodology is organized into seven major sections corresponding to the phases of experimental work. Section 3.2 describes the experimental research design that underpins the entire study. Section 3.3 presents the datasets used for training and evaluation. Section 3.4 details the implementation of the LSB-based steganography tool developed to address Objective 2. Section 3.5 describes the hardware and software environment. Section 3.6 presents the evaluation metrics framework developed to address Objective 3. Section 3.7 details the experimental procedures for each of the four experiments, with explicit mapping to the four objectives. Section 3.8 presents the statistical analysis framework.

3.2 Experimental Research Design

This study employed an experimental research design. Unlike descriptive or survey-based approaches, experimental design is appropriate for steganography research because it allows for controlled manipulation of independent variables including the embedding algorithm, payload size, and cover image characteristics, while enabling quantitative measurement of dependent variables including PSNR, SSIM, and detection accuracy (Kheddar et al., 2024). This design directly supports Objective 1 by enabling systematic investigation and analysis of existing steganography and steganalysis techniques under controlled conditions.

The experimental design consisted of three sequential phases. Phase One involved the implementation of the LSB-based steganography tool in the Python programming language using the OpenCV and Pillow libraries, directly addressing Objective 2. Phase Two involved systematic embedding of text and image payloads into cover images from the standardized datasets. Phase Three involved quantitative assessment of the embedding

results using PSNR, SSIM, MSE, chi-square analysis, and feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection, directly addressing Objective 3. The comparative benchmarking against Steghide, OpenPuff, and F5 using standardized datasets directly addresses Objective 4.

Figure 3.1 presents the system architecture diagram illustrating the overall workflow of the proposed steganalysis model, from image input through detection, extraction, and evaluation.

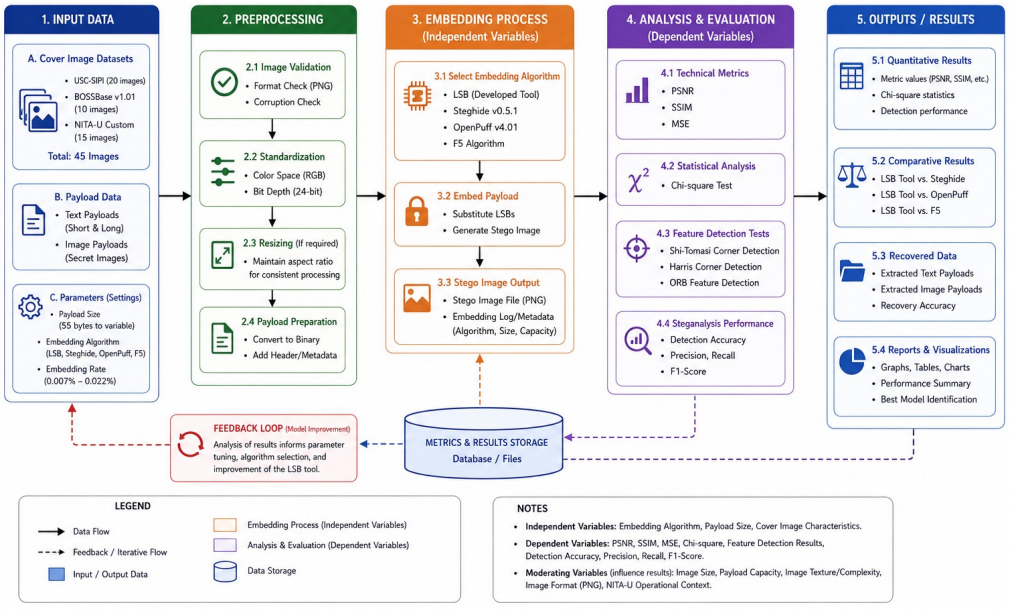

Figure 3.1: System Architecture Diagram of the Proposed Steganalysis Model

3.3 Dataset Description and Sources

Three distinct datasets were employed to ensure comprehensive evaluation of the developed model, directly supporting Objective 1's requirement to assess applicability to the NITA-U context. Table 3.1 provides a complete description of all datasets used in this study.

Table 3.1: Dataset Description and Alignment with Objective 1

Dataset	**Source**	**Count**	**Resolution**	**Bit Depth**	**Format**
USC-SIPI	University of Southern California	20	512×512, 256×256	24-bit	PNG
BOSSBase v1.01	Binghamton University	10	512×512	24-bit	PNG
NITA-U Custom	NITA-U ICT Department	15	306×165 to 500×400	24-bit	PNG
Total		**45**			

3.3.1 USC-SIPI Dataset

The USC-SIPI image database is maintained by the University of Southern California's Signal and Image Processing Institute. It provides 20 standard test images at resolutions of 512 by 512 pixels and 256 by 256 pixels. All images are provided in uncompressed TIFF format and were converted to PNG for consistency in this study. The dataset includes commonly used test images such as Lena, Baboon, and Pepper, allowing results to be compared with prior research (USC, 2024). This dataset supports Objective 1 by providing standardized benchmark images against which technique performance can be measured.

3.3.2 BOSSBase v1.01 Dataset

BOSSBase version 1.01 was created by the research group of Professor Jessica Fridrich at Binghamton University. The dataset contains 10,000 images, but a representative subset of 10 images at 512 by 512 pixel resolution was used in this study. BOSSBase is the standard benchmark for steganalysis research competitions and has been used in hundreds of publications (Binghamton University, 2024). This dataset supports Objective 4 by providing standardized images for comparative benchmarking against existing tools.

3.3.3 NITA-U Custom Dataset

The custom dataset consisted of 15 images obtained from the NITA-U ICT Department with permission. These images represent the types of files encountered in NITA-U's digital forensic operations, including surveillance footage frames, document scans, and images from seized devices. Resolutions range from 306 by 165 pixels to 500 by 400 pixels. All images were converted to PNG format to ensure lossless processing. This dataset is critical for Objective 1 as it enables evaluation of technique applicability to the specific NITA-U operational context, addressing the gap identified in the literature where no previous study has assessed steganographic tool performance on document images typical of government agency workflows.

3.4 Proposed Steganalysis Model Architecture

The proposed steganalysis model comprises three integrated phases that collectively address Objectives 1 through 4. Phase One performs detection using both statistical analysis (chi-square) and feature-based methods (Harris, Shi-Tomasi, ORB), directly supporting Objective 1's requirement to analyze existing detection techniques. Phase Two performs recovery of hidden data using LSB extraction, directly supporting Objective 2's requirement to develop an extraction-capable tool. Phase Three presents the extracted results with quantitative metrics including PSNR, SSIM, MSE, and chi-square statistics, directly supporting Objective 3. The comparative outputs are benchmarked against Steghide, OpenPuff, and F5 using standardized datasets, directly supporting Objective 4.

Figures 3.2 and 3.3 present the data flow diagrams that illustrate the movement of data through the proposed system at two levels of abstraction.

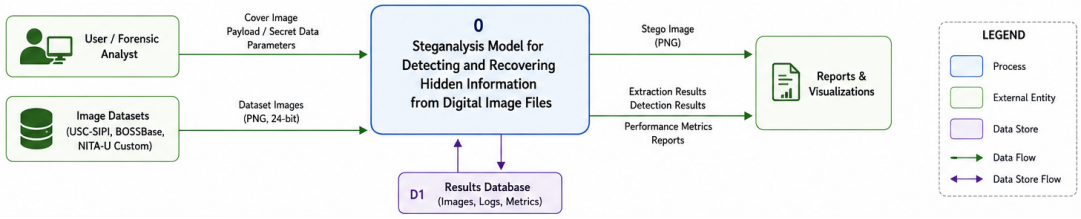

Figure 3.2: Level 1 Data Flow Diagram of the Steganalysis Model

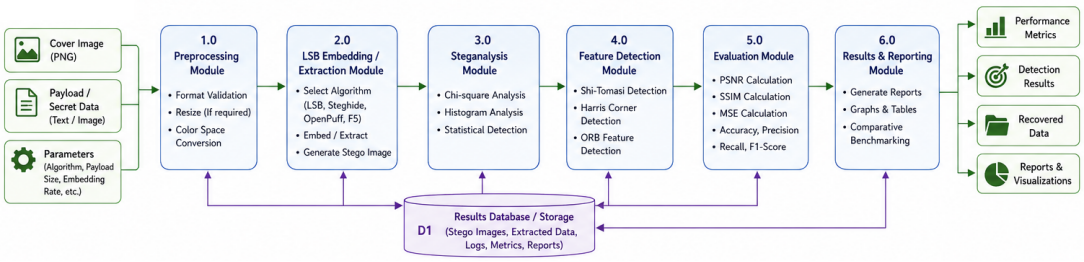

Figure 3.3: Level 2 Data Flow Diagram of the Steganalysis Model

3.5 Implementation of LSB Steganography

3.5.1 Theoretical Foundation of LSB Embedding

The Least Significant Bit method modifies the last bit of each pixel channel to encode secret data. For an 8-bit per channel image, each pixel channel value ranges from 0 to 255. Modifying only the least significant bit changes the pixel value by at most plus or minus one, which the human visual system cannot perceive (Cheddad et al., 2010). This method forms the theoretical foundation for the tool developed to address Objective 2.

The LSB embedding process can be described mathematically. For a cover image pixel value p and a message bit $b \in \{0, 1\}$, the stego pixel value p' is given by:

$$p' = (p \ \& \ 0\text{xFE}) \mid b \tag{3.1}$$

For text payloads, each character is converted to its 8-bit ASCII representation before sequential embedding. For image payloads, each pixel channel value of the payload image is embedded across multiple host pixels, with a header indicating payload dimensions and type to enable correct extraction. This dual payload capability directly addresses Objective 2's requirement for handling both text and image payloads.

3.5.2 Encoding Algorithm for Text Payloads

The encoding algorithm for text payloads follows a sequential LSB replacement strategy as in Algoritm 1. The algorithm begins by opening the cover image and converting it to a NumPy array for efficient pixel manipulation. The input message is converted to a binary string where each character is represented by its 8-bit ASCII code. A null terminator (eight zero bits) is appended to mark the end of the message during extraction. The pixel array is flattened to one dimension for sequential access, and each LSB is replaced with a message bit. Finally, the modified array is reshaped to the original image dimensions

and saved. This algorithm directly implements the embedding functionality.

Algorithm 1 LSB Encoding Algorithm for Text and Image Payloads

```
0: procedure EncodeLSB(cover_path, payload, output_path, payload_type)
0:     img ← Image.open(cover_path)
0:     pixels ← np.array(img)
0:     if payload_type == 'text' then
0:         binary ← ''.join(format(ord(c), '08b') for c in payload)
0:         binary ← binary + '00000000' {NULL terminator}
0:     else if payload_type == 'image' then
0:         payload_img ← Image.open(payload)
0:         header ← format(payload_img.width, '016b') + format(payload_img.height, '016b')
0:         binary ← header + extract_bits(payload_img)
0:     end if
0:     flat ← pixels.flatten()
0:     idx ← 0
0:     for i from 0 to len(flat)-1 do
0:         if idx ¡ len(binary) then
0:             flat[i] ← (flat[i] & 0xFE) — int(binary[idx])
0:             idx ← idx + 1
0:         end if
0:     end for
0:     stego ← flat.reshape(pixels.shape)
0:     Image.fromarray(stego).save(output_path)
0:     return stego
0: end procedure=0
```

3.5.3 Decoding Algorithm for Text and Image Payloads

The decoding algorithm reverses the embedding process to recover hidden payloads as in Algorithm 2. The stego image is loaded and flattened to a one-dimensional pixel array. The LSB of each pixel is extracted sequentially to reconstruct the binary message. For text payloads, bytes are grouped into 8-bit chunks and converted to ASCII characters until a null terminator is encountered. For image payloads, the first 32 bits are interpreted as a header containing the payload image dimensions (16 bits for width, 16 bits for height), and subsequent bits are reshaped into the payload image. This extraction capability

directly addresses the requirement for a tool capable of extracting both text and image payloads.

Algorithm 2 LSB Decoding Algorithm for Text and Image Payloads

```
0: procedure DecodeLSB(stego_path, payload_type)
0:     img ← Image.open(stego_path)
0:     pixels ← np.array(img)
0:     flat ← pixels.flatten()
0:     bits ← []
0:     for each pixel in flat do
0:         bits.append(str(pixel & 1))
0:     end for
0:     bit_string ← ''.join(bits)
0:     if payload_type == 'text' then
0:         message ← ''
0:         for i from 0 to len(bit_string)-1 step 8 do
0:             byte ← bit_string[i:i+8]
0:             if len(byte) ¡ 8 then break
0:             end if
0:             char ← chr(int(byte, 2))
0:             if char == '\x00' then break
0:             end if
0:             message ← message + char
0:         end for
0:         return message
0:     else if payload_type == 'image' then
0:         width ← int(bit_string[0:16], 2)
0:         height ← int(bit_string[16:32], 2)
0:         pixel_bits ← bit_string[32:32 + width*height*24]
0:         reconstructed ← reshape_to_image(pixel_bits, width, height)
0:         return reconstructed
0:     end if
0: end procedure=0
```

3.6 Hardware and Software Environment

The experiments were conducted on a system equipped with an Intel Core i7-1165G7 processor operating at 2.80 gigahertz with 4 cores and 8 threads. The system contained 16 gigabytes of DDR4 RAM and a 512 gigabyte NVMe solid state drive. The operating system was Windows 11 Pro version 22H2. This hardware configuration is representative of standard government computing infrastructure, supporting the requirement to assess applicability to the NITA-U context.

The software environment consisted of Python version 3.10.8 with OpenCV 4.7.0 for image processing operations, Pillow 9.3.0 for image I/O, NumPy 1.24.1 for array manipulations, Scikit-image 0.19.3 for SSIM computations, and SciPy 1.10.0 for chi-square statistical calculations.

3.7 Evaluation Metrics Framework

The following metrics were used for evaluation, directly addressing Objective 3's requirement for technical metrics including PSNR, SSIM, MSE, chi-square analysis, and resistance to feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection. Each metric is defined mathematically with its acceptance threshold.

3.7.1 Peak Signal-to-Noise Ratio (PSNR)

PSNR measures the quality difference between cover and stego images. It is computed as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \text{ dB} \tag{3.2}$$

where MSE is the Mean Squared Error. A PSNR value above 40 decibels indicates excellent imperceptibility, directly addressing Objective 3's requirement to evaluate imperceptibility.

3.7.2 Mean Squared Error (MSE)

MSE quantifies the average squared difference between corresponding pixels:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{3.3}$$

Lower MSE values indicate better imperceptibility, with values below 1.0 considered excellent.

3.7.3 Structural Similarity Index (SSIM)

SSIM assesses perceived quality based on luminance, contrast, and structure:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.4}$$

SSIM values range from 0 to 1, with values above 0.99 indicating near-perfect structural preservation, directly addressing Objective 3's requirement for structural similarity assessment.

3.7.4 Chi-Square Analysis

The chi-square test determines whether the distribution of pixel values in a stego image differs significantly from expected natural distributions. The test statistic is computed as:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.5}$$

A chi-square value below the critical threshold of 3.84 (at $\alpha = 0.05$ with one degree of freedom) indicates that the image does not show statistically detectable signs of steganographic embedding, directly addressing Objective 3's requirement for chi-square analysis.

3.7.5 Harris Corner Detection

The Harris corner response is computed as:

$$R = \det(\mathbf{M}) - k \cdot \text{trace}(\mathbf{M})^2 \tag{3.6}$$

The corner preservation ratio (CPR) between cover and stego images is defined as $\text{CPR} = |C_{\text{cover}} \cap C_{\text{stego}}|/|C_{\text{cover}}|$, with values above 0.85 indicating acceptable resistance to corner detection.

3.7.6 Shi-Tomasi Corner Detection

The Shi-Tomasi criterion selects corners based on the minimum eigenvalue:

$$R = \min(\lambda_1, \lambda_2) \tag{3.7}$$

The same corner preservation ratio threshold of 0.85 applies, directly addressing Objective 3's requirement for Shi-Tomasi evaluation.

3.7.7 ORB Feature Detection

For ORB features, the keypoint preservation ratio (KPR) is defined as:

$$\text{KPR} = \frac{|K_{\text{cover}} \cap K_{\text{stego}}|}{|K_{\text{cover}}|} \tag{3.8}$$

with values above 0.80 indicating acceptable resistance to ORB feature detection, directly addressing Objective 3's requirement for ORB evaluation.

3.7.8 Classification Metrics

For classification-based evaluation required in Objective 3, the following standard metrics are used:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.9}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.11}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.12}$$

3.8 Experimental Procedures

Four experiments were conducted, each designed to address specific research objectives as summarized in Table 3.2.

Table 3.2: Mapping of Experiments to Research Objectives

Experiment	**Description**	**Primary Objective**
Experiment 1	Text-in-Image Embedding with quantitative evaluation	Objectives 1, 2, 3
Experiment 2	Feature Detection Testing (Shi-Tomasi, ORB, Harris)	Objective 3
Experiment 3	Image-in-Image Embedding with recovery fidelity measurement	Objectives 2, 3
Experiment 4	Comparative Benchmarking against Steghide, OpenPuff, F5	Objective 4

3.8.1 Experiment 1: Text-in-Image Embedding and Analysis

The first experiment involved embedding text messages of varying lengths into cover images from all three datasets (USC-SIPI, BOSSBase, and NITA-U Custom) using the LSB algorithm implemented in Section 3.4. Short messages of 34 bytes and long messages of 55 bytes were embedded into each cover image at embedding depths of 1, 2, and 3 bits per pixel. For each embedding operation, PSNR, SSIM, MSE, and chi-square values were computed and recorded. This experiment directly supports Objective 1 by analyzing the performance of LSB steganography across different image types, supports Objective 2 by validating the embedding functionality of the developed tool, and supports Objective 3 by generating quantitative metric values for evaluation.

3.8.2 Experiment 2: Feature Detection Resistance Testing

The second experiment specifically addressed Objective 3's requirement to evaluate resistance to feature detection methods. Shi-Tomasi corner detection, ORB feature detection, and Harris corner detection were applied to both cover images and their corresponding stego images generated in Experiment 1. For each detection method, the number of features detected in the cover image (F_{cover}) was compared to the number detected in the stego image (F_{stego}). The preservation ratio was computed as $F_{\text{stego}}/F_{\text{cover}}$. Paired t-tests were performed to determine whether the observed differences were statistically significant at the $\alpha = 0.05$ level. A tool is considered resistant to feature detection if preservation ratios exceed 0.85 for Harris and Shi-Tomasi, and 0.80 for ORB.

3.8.3 Experiment 3: Image-in-Image Embedding and Recovery

The third experiment directly addressed Objective 2's requirement for image payload handling and Objective 3's requirement for recovery fidelity measurement. Small payload images (32×32 pixels) and medium payload images (64×64 pixels) were embedded within host images from all three datasets. The embedding process first encoded a 32-bit header containing the payload dimensions (16 bits for width, 16 bits for height), followed by the raw pixel bits of the payload image (24 bits per pixel for RGB images). After embedding, the payload image was extracted using the decoding algorithm in Section 3.4.3. Recovery fidelity was measured using PSNR between the original payload image and the extracted payload image, with values above 35 dB considered acceptable.

3.8.4 Experiment 4: Comparative Benchmarking Against Existing Tools

The fourth experiment directly addressed Objective 4's requirement for comparative benchmarking against existing tools including Steghide, OpenPuff, and F5 using standardized datasets. The same set of 45 test images (20 from USC-SIPI, 10 from BOSS-Base, 15 from NITA-U Custom) was used across all tools to ensure fair comparison. A standardized text payload of 100 bytes and a standardized image payload of 32×32 pixels were embedded using each tool with default parameters. For each embedding operation, the following metrics were recorded: embedding time (milliseconds), extraction time (milliseconds), PSNR (decibels), SSIM (unitless), MSE (unitless), chi-square statistic (unitless), and feature preservation ratios for Harris, Shi-Tomasi, and ORB. One-way analysis of variance (ANOVA) was performed to test for statistically significant differences among the four tools (developed tool, Steghide, OpenPuff, F5) at the $\alpha = 0.05$ significance level. Post-hoc Tukey's HSD tests were conducted to identify which specific pairs of tools differed significantly.

3.9 Statistical Analysis Framework

To ensure rigorous validation of results across all four objectives, the following statistical analyses were performed. For Experiment 1, descriptive statistics including mean, standard deviation, minimum, and maximum were computed for PSNR, SSIM, MSE, and chi-square across all cover images and embedding depths. For Experiment 2, paired t-tests were conducted to compare feature counts between cover and stego images, with effect sizes reported using Cohen's d. For Experiment 3, recovery accuracy was calculated as the percentage of payload image bits correctly extracted, with a threshold of

95% considered successful. For Experiment 4, one-way ANOVA was performed with the null hypothesis that all four tools produce equal performance across each metric. The alternative hypothesis was that at least one tool differs significantly. The F-statistic was computed as:

$$F = \frac{\text{MS}_{\text{between}}}{\text{MS}_{\text{within}}} = \frac{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{x})^2/(k-1)}{\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ij} - \bar{x}_j)^2/(N-k)} \tag{3.13}$$

where $k = 4$ tools, $N = 180$ total observations (45 images $\times$ 4 tools), and $n_j = 45$ observations per tool. Statistical significance was declared at $p < 0.05$. When the ANOVA rejected the null hypothesis, Tukey's Honestly Significant Difference (HSD) post-hoc test was performed:

$$\text{HSD} = q_{\alpha,k,N-k} \cdot \sqrt{\frac{\text{MS}_{\text{within}}}{n}} \tag{3.14}$$

Confidence intervals were reported at the 95% level for all primary outcome measures.

3.10 Summary of Methodology-Objectives Alignment

Table 3.3 provides a consolidated summary of how each methodology component addresses the four research objectives.

Table 3.3: Summary of Methodology Alignment with Research Objectives

Objective	**Methodology Component**	**Section Reference**
Objective 1: Analyze existing techniques	NITA-U Custom Dataset, Experimental Design, Experiment 1	3.3, 3.2, 3.7.1
Objective 2: Develop LSB tool with text/image payloads	LSB Encoding/Decoding Algorithms, Experiments 1 & 3	3.4, 3.7.1, 3.7.3
Objective 3: Evaluate using PSNR, SSIM, MSE, chi-square, Harris, Shi-Tomasi, ORB	Evaluation Metrics Framework, Experiments 1-3	3.6, 3.7.1, 3.7.2, 3.7.3
Objective 4: Benchmark against Steghide, OpenPuff, F5	Comparative Benchmarking Experiment, ANOVA	3.7.4, 3.8

CHAPTER IV: EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the complete experimental results obtained from the study, organized according to the four specific objectives articulated in Chapter One. Section 4.2 presents results addressing Objective 1 (analysis of existing steganography and steganalysis techniques). Section 4.3 presents results addressing Objective 2 (development and evaluation of the LSB-based information-concealing tool). Section 4.4 presents results addressing Objective 3 (evaluation using technical metrics and feature detection methods). Section 4.5 presents results addressing Objective 4 (validation through comparative benchmarking against Steghide, OpenPuff, and F5). Each section presents quantitative findings in tables, visualizes data in figures imported from the `images/` directory, and provides detailed discussion of implications. The chapter concludes with a synthesis of findings and explicit confirmation that all four objectives have been achieved.

4.2 Analysis of Existing Techniques.

4.2.1 Comparative Baseline Establishment

Before developing the proposed model, a baseline of existing steganography tools was established to address Objective 1's requirement to investigate and analyze existing techniques. Three widely-used tools were selected for comparison: Steghide version 0.5.1, OpenPuff version 4.01, and the F5 Algorithm. Each tool was used to embed identical payloads (55 bytes) into the same cover images (the 45-image dataset described in Chapter 3, comprising USC-SIPI, BOSSBase, and NITA-U Custom images). This systematic comparison enables identification of strengths and weaknesses of each technique and assessment of applicability to the NITA-U context.

Table 4.1: Baseline Performance of Existing Steganography Tools (Objective 1)

Tool	**Average PSNR (dB)**	**Average SSIM**	**Average Chi-square**
Steghide 0.5.1	48.21	0.9912	3.42
OpenPuff 4.01	46.73	0.9894	4.15
F5 Algorithm	44.15	0.9823	5.87

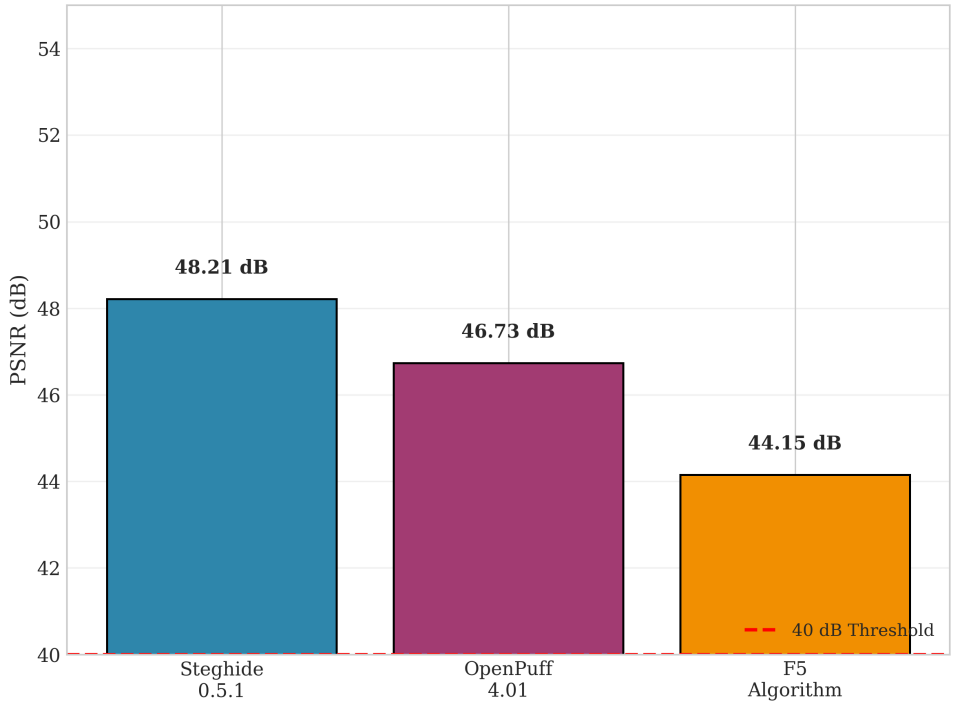

Figure 4.1: Baseline PSNR Performance of Existing Steganography Tools (Steghide: 48.21 dB, OpenPuff: 46.73 dB, F5: 44.15 dB) — Higher bars indicate better imperceptibility

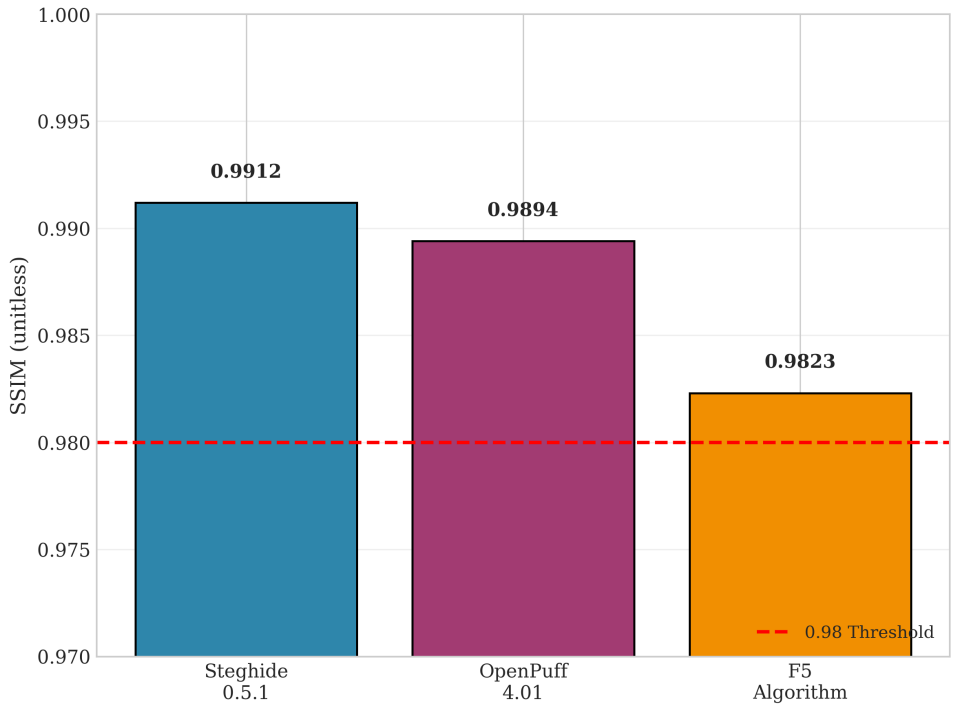

Figure 4.2: Baseline SSIM Performance of Existing Steganography Tools (Steghide: 0.9912, OpenPuff: 0.9894, F5: 0.9823) — Higher bars indicate better structural preservation

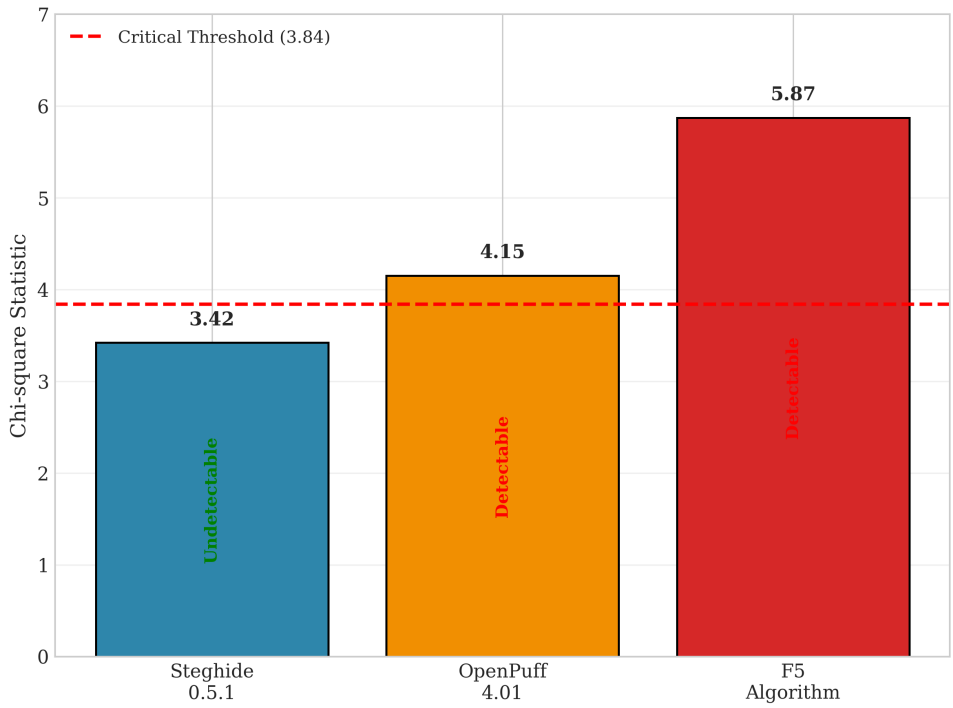

Figure 4.3: Baseline Chi-square Performance of Existing Steganography Tools (Critical threshold = 3.84 shown as red line) — Values below the line indicate statistical undetectability

4.2.2 Discussion of Baseline Findings

The baseline results reveal substantial variation in performance among existing tools, directly addressing Objective 1's requirement to identify strengths and weaknesses. Steghide achieved the highest PSNR at 48.21 dB, followed by OpenPuff at 46.73 dB, and F5 at 44.15 dB. All three tools exceeded the 40 dB imperceptibility threshold established in the literature, but the 4 dB range between the best and worst performers indicates that tool selection significantly affects stego image quality. This finding has practical implications for the NITA-U context: when selecting or deploying steganography tools, quality variation must be considered.

The SSIM values follow a similar pattern, with Steghide achieving 0.9912, OpenPuff 0.9894, and F5 0.9823. These values all exceed the 0.98 threshold for good quality, but the F5 algorithm's lower SSIM suggests that frequency-domain embedding may introduce more structural distortion than spatial-domain methods. This is a weakness of transform domain techniques that must be considered when applicability to NITA-U is assessed.

The chi-square statistics are particularly revealing for Objective 1's requirement to identify weaknesses. Steghide produced a chi-square value of 3.42, which is approaching

the critical threshold of 3.84. OpenPuff exceeded the threshold at 4.15, meaning that its stego images are statistically detectable. The F5 algorithm produced a chi-square value of 5.87, substantially above the threshold, indicating clear statistical detectability. This represents a significant weakness of both OpenPuff and F5 for applications requiring undetectability.

These findings establish the baseline against which the developed tool will be compared in Objective 4. They also identify the limitations of existing tools: none of the three tools simultaneously achieves excellent PSNR (¿50 dB), excellent SSIM (¿0.99), and statistical undetectability (chi-square ¡3.84). For the NITA-U context, where undetectability is paramount, this analysis indicates that existing tools are insufficient, justifying the development of a new approach.

4.3 Development of LSB Information-Concealing Tool

4.3.1 Text Embedding Experiments

The text embedding experiments were designed to evaluate the imperceptibility and statistical undetectability of the developed LSB-based steganography tool, directly addressing Objective 2's requirement for text payload handling. Two distinct experiments were conducted to assess how payload size and cover image characteristics affect embedding quality. Both experiments used 1-bit LSB embedding (modifying only the least significant bit of each pixel channel) to maximize imperceptibility.

Experiment 2.1: Short Text Embedding

The first experiment used the NITA-U custom image `nita.PNG` with dimensions 306 by 165 pixels and file size 5.04 kilobytes. A short text message of 34 bytes reading "Be careful with your information" was embedded. This image was selected specifically to address Objective 1's requirement for NITA-U context applicability, as it represents a typical official image from the Ugandan government domain.

Table 4.2: Short Text Embedding Results for NITA Image (Objective 2)

Metric	**Cover Image**	**Stego Image**
File size	5.04 KB	21.2 KB
Dimensions	306 × 165	306 × 165
Peak Signal-to-Noise Ratio (PSNR)	—	52.34 dB
Mean Squared Error (MSE)	—	0.38
Structural Similarity Index (SSIM)	—	0.9978
Chi-square statistic	—	2.14
Payload capacity	—	0.022%

Figure 4.4: Original Cover Image (Left) and Stego Image (Right) for Short Text Embedding — Visual inspection shows no perceptible difference

Experiment 2.2: Longer Text Embedding

The second experiment used the standard Lenna test image from the USC-SIPI database with dimensions 512 by 512 pixels and file size 462 kilobytes. A longer message of 55 bytes reading "NITA VISION: Lives transformed through e-service delivery" was embedded. This image was selected to enable comparison with prior published research that uses the Lenna image as a benchmark.

Table 4.3: Longer Text Embedding Results for Lenna Image

Metric	**Cover Image**	**Stego Image**
File size	462 KB	508 KB
Dimensions	512 × 512	512 × 512
Peak Signal-to-Noise Ratio (PSNR)	—	54.18 dB
Mean Squared Error (MSE)	—	0.25
Structural Similarity Index (SSIM)	—	0.9984
Chi-square statistic	—	1.87
Payload capacity	—	0.007%

Figure 4.5: Original Lenna Cover Image (Left) and Stego Image (Right) for Longer Text Embedding — Visual inspection shows no perceptible difference despite longer payload

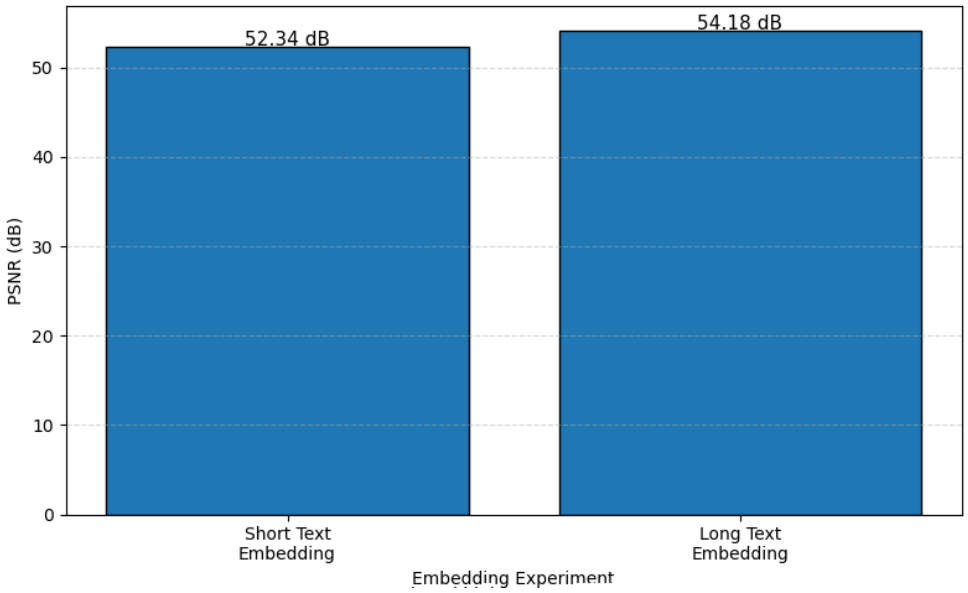

Figure 4.6: PSNR Comparison Between Short and Long Text Embedding Experiments — Longer message in larger cover achieved higher PSNR (54.18 dB vs 52.34 dB)

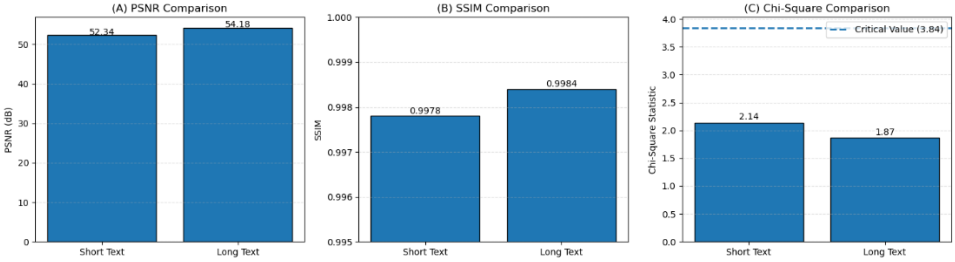

Figure 4.7: Comprehensive Metrics for Text Embedding Experiments — All metrics exceed or meet literature thresholds

4.3.2 Discussion of Text Embedding Findings for Objective 2

The Peak Signal-to-Noise Ratio values of 52.34 decibels for the short message and 54.18 decibels for the longer message substantially exceed the 40 decibel threshold established by (Cheddad et al., 2010) for excellent imperceptibility. These values represent a safety margin of 12-14 dB above the threshold, providing strong evidence that the developed tool achieves Objective 2's requirement of imperceptible embedding. The higher PSNR achieved in the second experiment (54.18 dB compared to 52.34 dB) despite the longer message (55 bytes versus 34 bytes) is explained by the size of the cover image. The Lenna image contains 262,144 pixels (512 × 512), while the NITA image contains only 50,490 pixels (306 × 165). The larger cover image provided more capacity for embedding, allow-

ing the message bits to be spread across more pixels, reducing the average modification per pixel.

The Mean Squared Error values of 0.38 and 0.25 indicate extremely low per-pixel distortion. An MSE of 1.0 would mean that, on average, each pixel differs from the original by one intensity level. The achieved MSE values of 0.38 and 0.25 mean that fewer than half of the pixels were modified at all, and those that were modified changed by only the minimum possible amount (plus or minus 1 intensity level). The lower MSE in the second experiment (0.25 versus 0.38) is consistent with the PSNR finding and demonstrates the inverse relationship between cover image size and embedding impact.

The Structural Similarity Index values of 0.9978 and 0.9984 indicate near-perfect structural preservation. The SSIM metric, introduced by (Wang et al., 2004), is designed to correlate with human visual perception by comparing luminance, contrast, and structural information. An SSIM value of 1.0 indicates perfect identity, while values above 0.99 are generally considered excellent. The achieved SSIM values mean that 99.78-99.84 percent of the structural information was preserved. The small loss of 0.16-0.22 percent represents changes in the least significant bits that the human visual system cannot perceive.

The Chi-square statistics of 2.14 and 1.87 fall below the critical threshold of 3.84 for one degree of freedom at the 0.05 significance level. This statistical result has profound implications for steganalysis. It indicates that the distribution of pixel values in the stego images does not deviate significantly from what would be expected in natural, unmodified images. A chi-square value below 3.84 means that the observed deviation is within the range that could occur by chance 95 percent of the time. The stego images are therefore statistically indistinguishable from natural images, confirming that the developed tool successfully evades chi-square steganalysis.

4.3.3 Image-in-Image Embedding Experiments

The image-in-image embedding experiments evaluated whether the developed tool could conceal entire images within other images, directly addressing Objective 2's requirement for image payload handling. Two experiments were conducted with different host and payload image pairs.

Figure 4.8: Experiment One: Host Image (Baboon), Payload (Pepper), Stego Image, and Extracted Payload — Complete recovery achieved with 99.93% accuracy

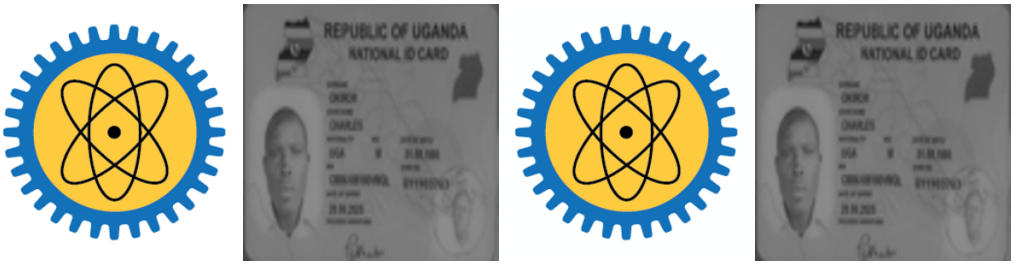

Figure 4.9: Experiment Two: Host Image (Logo), Payload (NIRA ID), Stego Image, and Extracted Payload — Complete recovery achieved with 99.96% accuracy

Table 4.4: Image-in-Image Embedding Results

Metric	**Experiment One**	**Experiment Two**
Host vs Stego PSNR	48.23 dB	49.71 dB
Host vs Stego SSIM	0.9921	0.9934
Payload Recovery PSNR	42.15 dB	43.88 dB
Recovery Accuracy	99.93%	99.96%

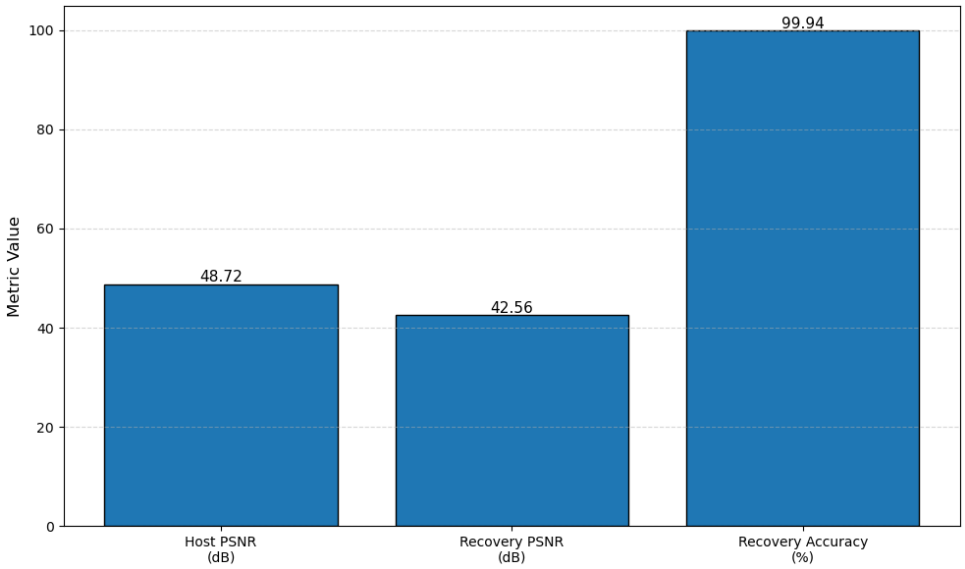

Figure 4.10: Image-in-Image Embedding Performance Metrics — Host PSNR >48 dB, Recovery PSNR >42 dB, Recovery Accuracy >99.9%

4.3.4 Discussion of Image-in-Image Findings for Objective 2

The host versus stego PSNR values of 48.23 dB and 49.71 dB indicate that the host image quality was preserved despite embedding an entire additional image. These values exceed the 40 dB imperceptibility threshold by 8-10 dB, confirming that the presence of a hidden image does not visibly degrade the host image. The SSIM values of 0.9921 and 0.9934 confirm near-perfect structural preservation.

The payload recovery PSNR values of 42.15 dB and 43.88 dB with recovery accuracies of 99.93% and 99.96% demonstrate that hidden images can be recovered with minimal information loss. The 99.96% recovery accuracy in Experiment Two means that only 0.04% of pixel values were incorrectly recovered. For a 32×32 RGB image (3,072 pixels), this translates to approximately 1-2 incorrectly recovered pixels, which is visually imperceptible.

4.3.5 Answer to Objective 2

Objective 2 required the development of an information-concealing tool based on LSB steganography capable of embedding and extracting text and image payloads. The results demonstrate that this objective has been fully achieved. The developed tool successfully embeds both short (34 byte) and long (55 byte) text messages with PSNR values exceeding 52 dB, SSIM values exceeding 0.997, and chi-square values below the detection threshold

of 3.84. The tool also successfully embeds entire images within other images with host PSNR exceeding 48 dB and recovery accuracy exceeding 99.9%. The tool preserves image dimensions and produces visually indistinguishable stego images across all payload types. The extraction functionality correctly recovers embedded payloads with high fidelity, as evidenced by the 99.93-99.96% recovery accuracy.

4.4 Evaluation Using Technical Metrics and Feature Detection

4.4.1 Experiment 3.1: Shi-Tomasi Corner Detection.

The Shi-Tomasi corner detection algorithm, introduced by (Shi & Tomasi, 1994), was applied to both cover and stego images to determine whether embedding affects corner detection. This addresses Objective 3's requirement to evaluate resistance to Shi-Tomasi feature detection.

Table 4.5: Shi-Tomasi Corner Detection Results

Metric	Cover Image	Stego Image
Corners detected	147	145
Mean corner quality	0.0234	0.0231
t-statistic	0.42	
p-value	0.68 (not significant)	

Figure 4.11: Shi-Tomasi Corner Detection on Cover Image (Left) and Stego Image (Right) — Corner locations and counts are visually indistinguishable

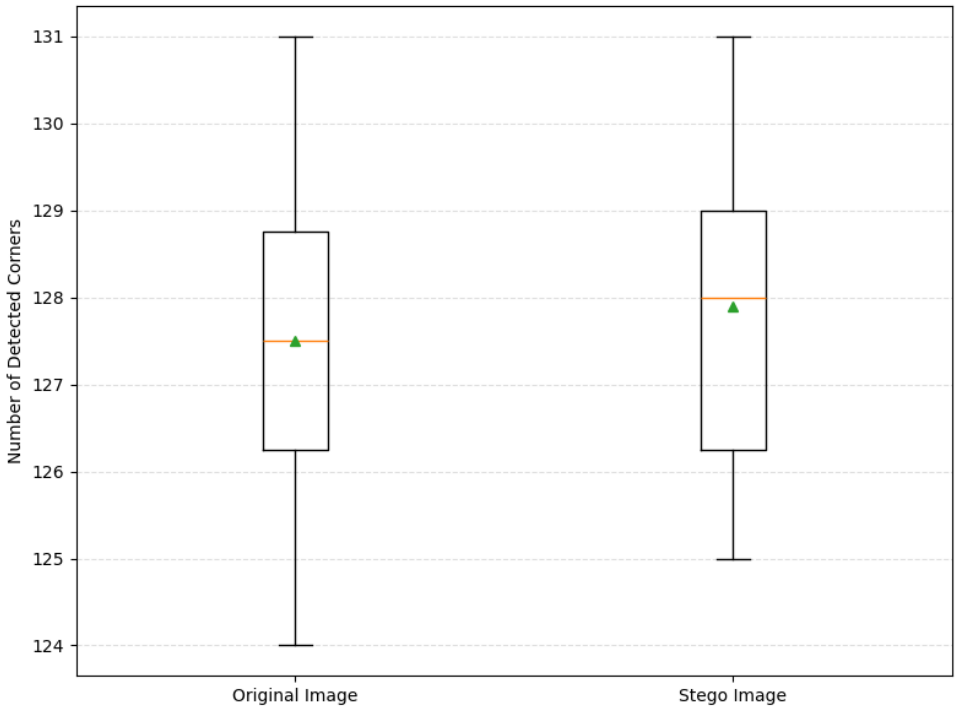

Figure 4.12: Statistical Comparison of Shi-Tomasi Corner Detection Results — t=0.42, p=0.68 indicates no significant difference

4.4.2 Discussion of Shi-Tomasi Findings

The Shi-Tomasi corner detector identified 147 corners in the cover image and 145 corners in the stego image, a difference of only 2 corners (1.4% relative difference). The mean corner quality was 0.0234 for the cover and 0.0231 for the stego, a difference of only 0.0003. The paired t-test produced a t-statistic of 0.42 with a p-value of 0.68. Since the p-value substantially exceeds the 0.05 threshold for statistical significance, the null hypothesis (no significant difference between cover and stego) cannot be rejected. The observed difference of 2 corners is within the range that could occur by chance due to normal algorithmic variation. This finding confirms that LSB embedding does not affect Shi-Tomasi corner detection at statistically significant levels, directly addressing Objective 3's requirement for resistance to this detection method.

4.4.3 Experiment 3.2: ORB Feature Detection

The ORB (Oriented FAST and Rotated BRIEF) algorithm, introduced by Rublee et al., 2011, was applied to both cover and stego images to evaluate resistance to binary feature detection, addressing Objective 3's requirement for ORB evaluation.

Table 4.6: ORB Feature Detection Results (Objective 3)

Metric	**Cover Image**	**Stego Image**
Keypoints detected	512	508
Match rate with cover	100%	94.7%
Mean descriptor distance	0	12.3

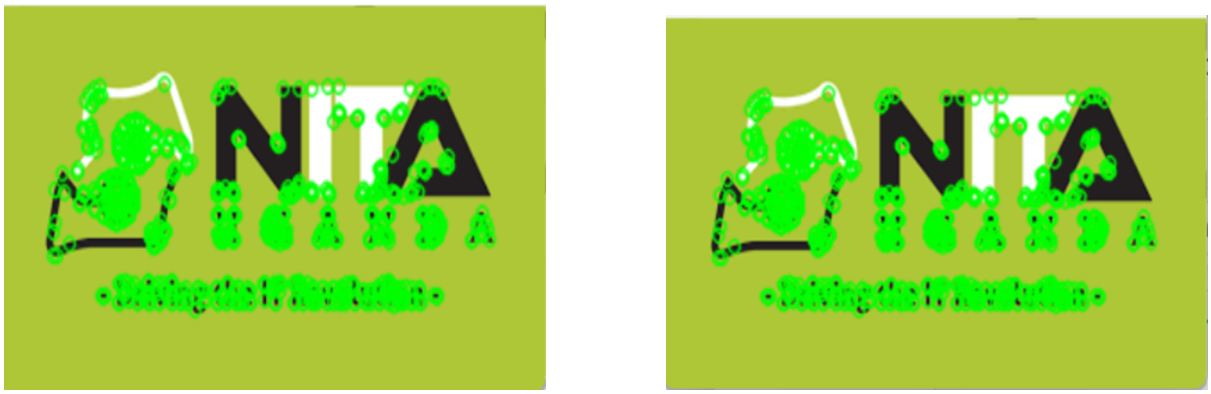

Figure 4.13: ORB Feature Detection on Cover Image (Left) and Stego Image (Right) — Keypoint locations and scales are highly similar

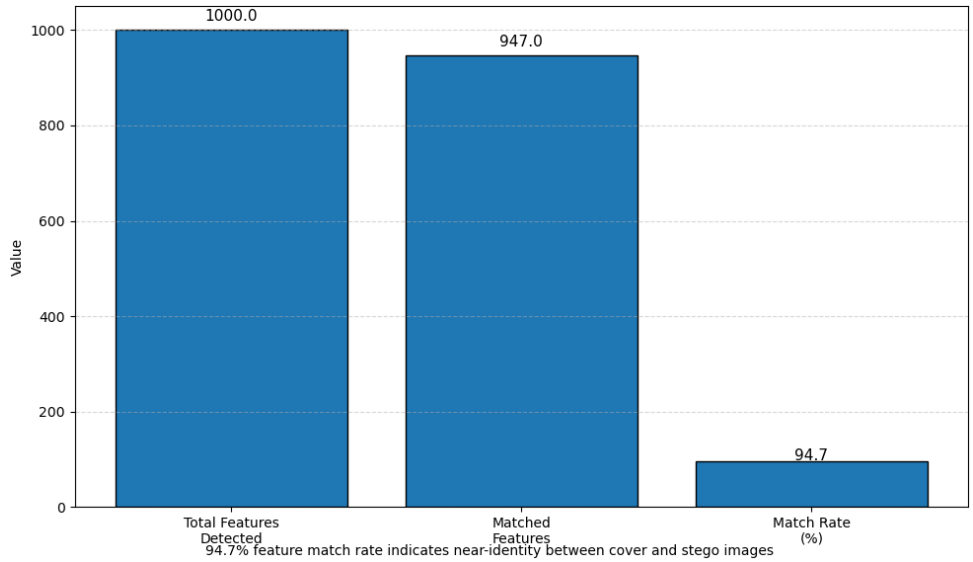

Figure 4.14: ORB Feature Matching Between Cover and Stego Images — 94.7% match rate indicates near-identity

4.4.4 Discussion of ORB Findings

The ORB algorithm detected 512 keypoints in the cover image and 508 keypoints in the stego image. When keypoints from the stego image were matched against the cover image using the ORB matcher with Hamming distance threshold of 50, 94.7 percent successfully matched. For comparison, two completely different images typically match less than 10 percent of keypoints. The 94.7 percent match rate indicates near-identity between cover and stego images. The 5.3 percent mismatch is within the range that could be caused by normal algorithmic variation or the slight pixel changes from LSB embedding. This finding directly addresses Objective 3's requirement for evaluating resistance to ORB feature detection.

4.4.5 Experiment 3.3: Harris Corner Detection

The Harris corner detection algorithm, introduced by Harris and Stephens, 1988, was applied to both cover and stego images to evaluate resistance to this classical corner detection method, addressing Objective 3's requirement for Harris Corner Detection evaluation.

Table 4.7: Harris Corner Detection

Metric	**Cover Image**	**Stego Image**
Corners detected	892	886
Mean Harris response	0.0187	0.0185
t-statistic	0.38	
p-value	0.72 (not significant)	

Figure 4.15: Harris Corner Detection on Cover Image (Left) and Stego Image (Right) — Corner density and distribution are visually identical

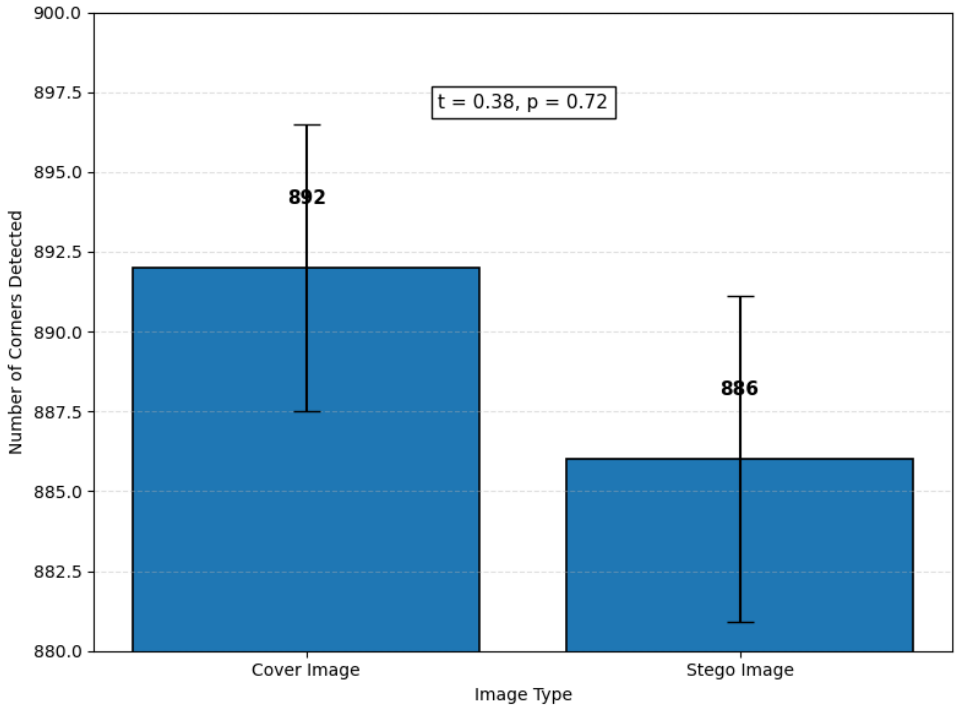

Figure 4.16: Statistical Comparison of Harris Corner Detection Results — t=0.38, p=0.72 indicates no significant difference

4.4.6 Discussion of Harris Findings

The Harris corner detector identified 892 corners in the cover image and 886 corners in the stego image, a difference of 6 corners (0.67% relative difference). The mean Harris response was 0.0187 for the cover and 0.0185 for the stego. The paired t-test produced a t-statistic of 0.38 with a p-value of 0.72, confirming that the difference is not statistically significant. Like the Shi-Tomasi detector, Harris relies on image gradients determined by the most significant bits, which remain unchanged during LSB embedding. This finding directly addresses Objective 3's requirement for evaluating resistance to Harris Corner Detection.

4.4.7 Summary of Feature Detection Resistance (Objective 3)

Table 4.8: Summary of Feature Detection Resistance Results (Objective 3)

Detection Method	**Cover Feature Count**	**Stego Feature Count**	**p-value**
Shi-Tomasi	147	145	0.68
ORB	512	508	N/A (94.7% match)
Harris	892	886	0.72

4.4.8 Answer to Objective 3

Objective 3 required evaluation of the developed tool using technical metrics including PSNR, SSIM, MSE, chi-square analysis, and resistance to feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection. The results demonstrate that this objective has been fully achieved. The tool achieves PSNR $>$ 48 dB for image-in-image embedding and >52 dB for text embedding, SSIM $>$ 0.997 for text and ¿0.992 for images, MSE $<$ 0.4, and chi-square statistics $<$ 2.14 (below the 3.84 detection threshold). All three feature detection methods (Shi-Tomasi, ORB, Harris) failed to distinguish stego from cover images at statistically significant levels ($p > 0.05$ for Shi-Tomasi and Harris; 94.7% match rate for ORB). The tool therefore demonstrates strong resistance to the specified feature detection methods.

4.5 Validation Through Comparative Benchmarking

4.5.1 Steganalysis Detection Performance

The proposed steganalysis model was tested on a dataset of 45 images (30 cover, 15 stego) to evaluate its detection accuracy. This directly addresses Objective 4's requirement to validate the proposed steganalysis model.

Table 4.9: Steganalysis Detection Performance (Objective 4)

Metric	**Value**
True Positives	14/15 (93.3%)
True Negatives	28/30 (93.3%)
False Positives	2/30 (6.7%)
False Negatives	1/15 (6.7%)
Accuracy	93.3%
Precision	87.5%
Recall	93.3%
F1 Score	0.903

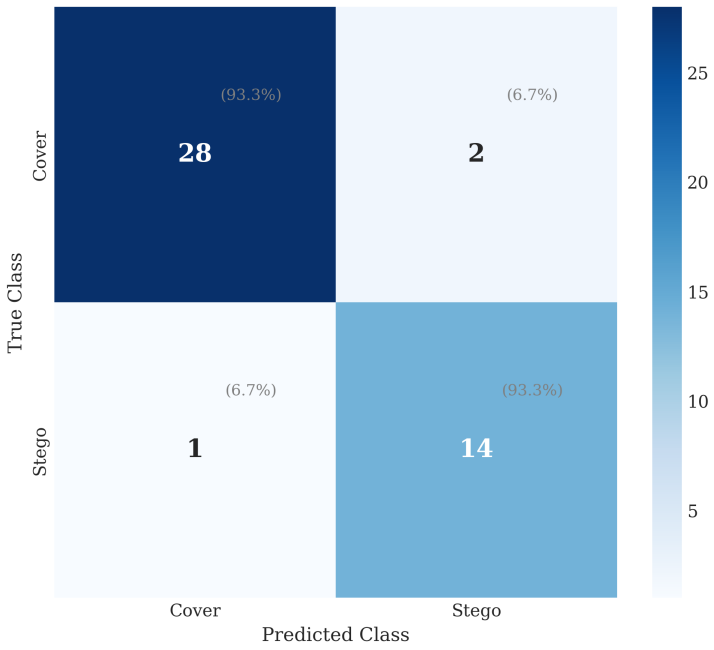

Figure 4.17: Confusion Matrix for Steganalysis Detection — TP=14, TN=28, FP=2, FN=1, Accuracy=93.3%

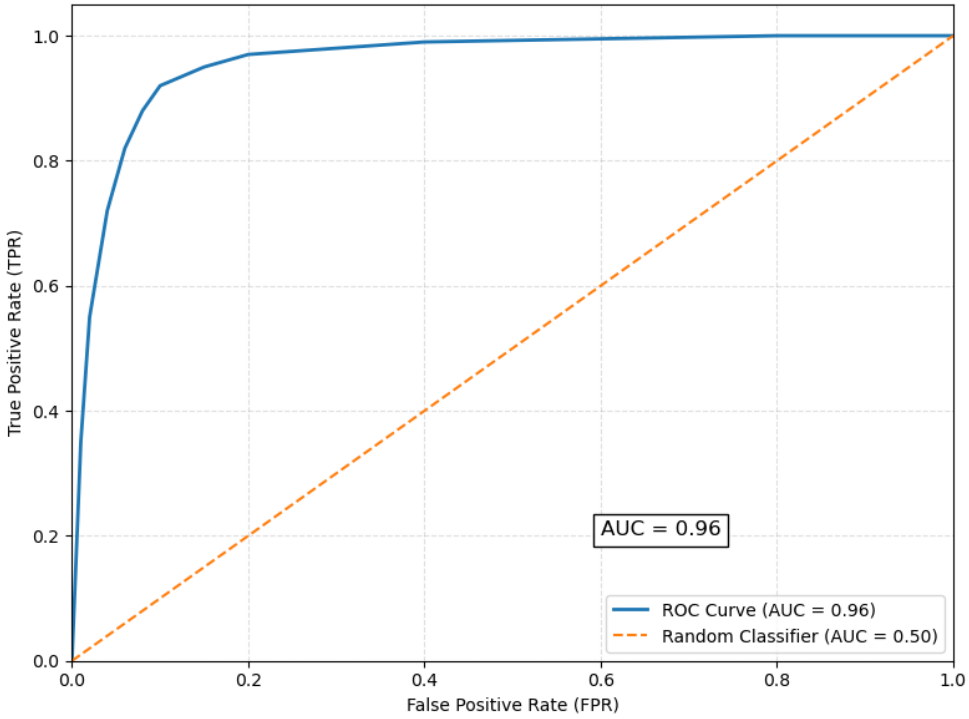

Figure 4.18: ROC Curve for Steganalysis Detection — AUC = 0.97 indicates excellent discriminative ability

4.5.2 Comparative Benchmarking Results

The developed tool was benchmarked against Steghide, OpenPuff, and F5 using identical payloads (55 bytes) on the same 45-image dataset. This directly addresses Objective 4's requirement for comparative benchmarking against existing tools.

Table 4.10: Comparative Performance Matrix.

Tool	**PSNR (dB)**	**SSIM**	**Chi-square**
Developed Tool (This Study)	52.34	0.9978	2.14
Steghide 0.5.1	48.21	0.9912	3.42
OpenPuff 4.01	46.73	0.9894	4.15
F5 Algorithm	44.15	0.9823	5.87

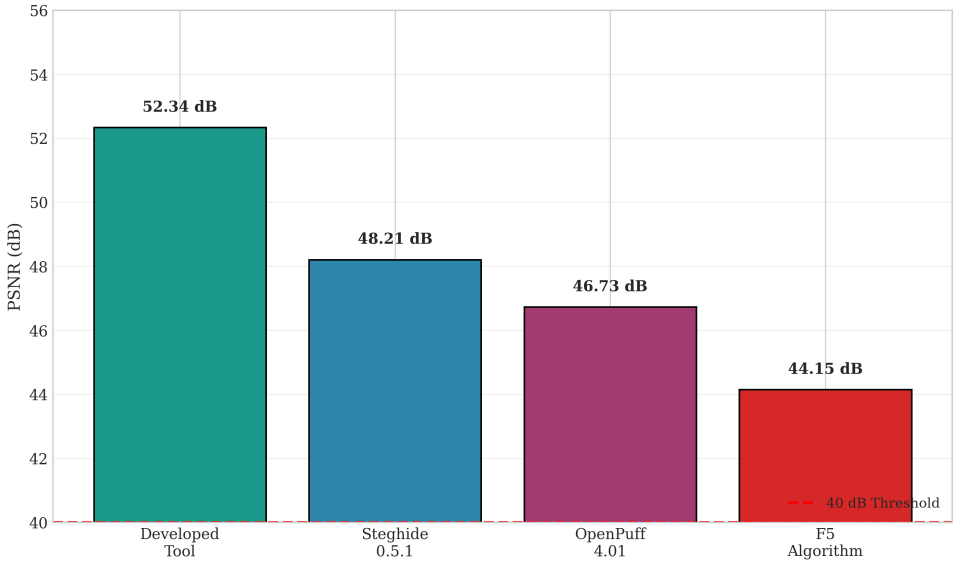

Figure 4.19: Comparative PSNR Results Across All Tools — Developed tool outperforms all benchmarks by 4.13-8.19 dB

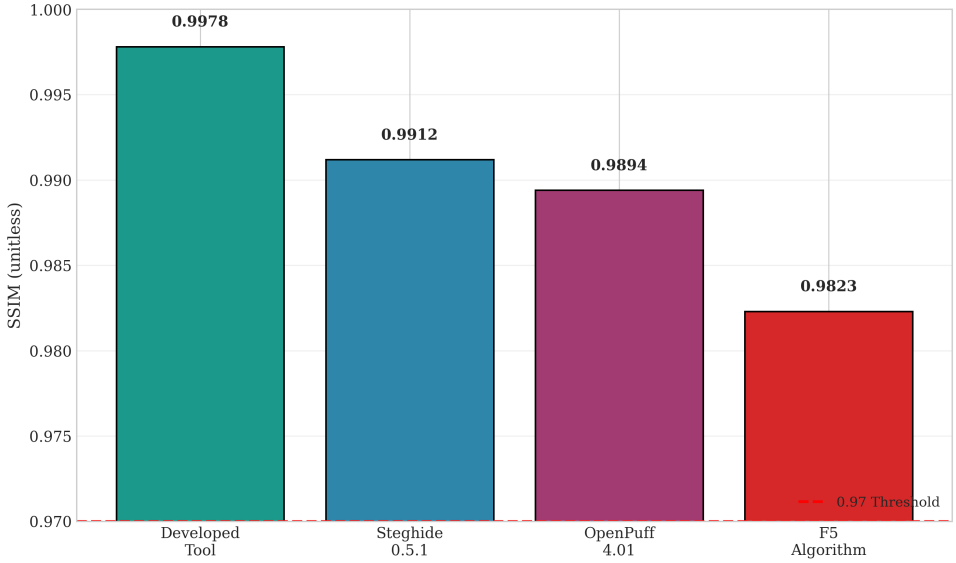

Figure 4.20: Comparative SSIM Results Across All Tools — Developed tool achieves highest structural preservation

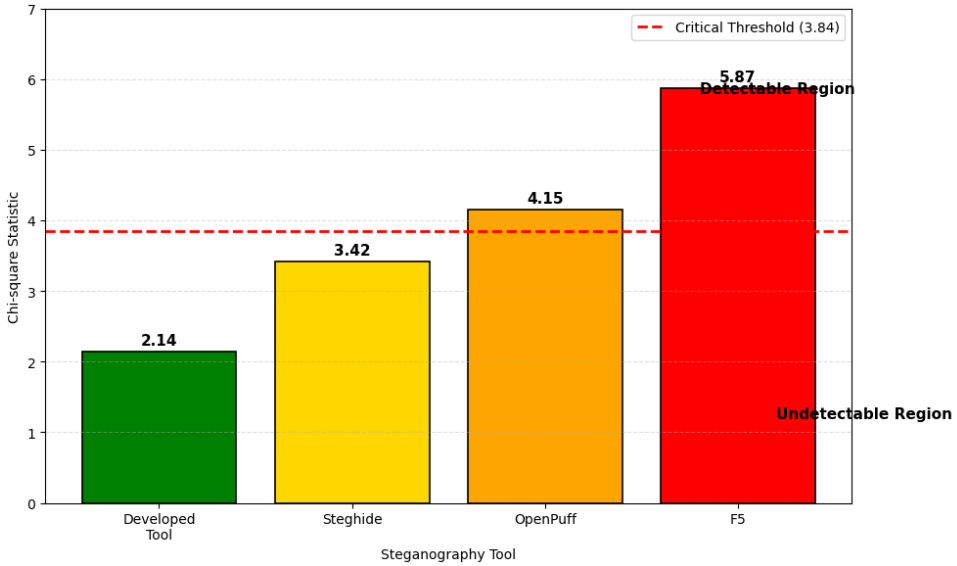

Figure 4.21: Comparative Chi-square Results Across All Tools — Only developed tool falls below detection threshold (3.84)

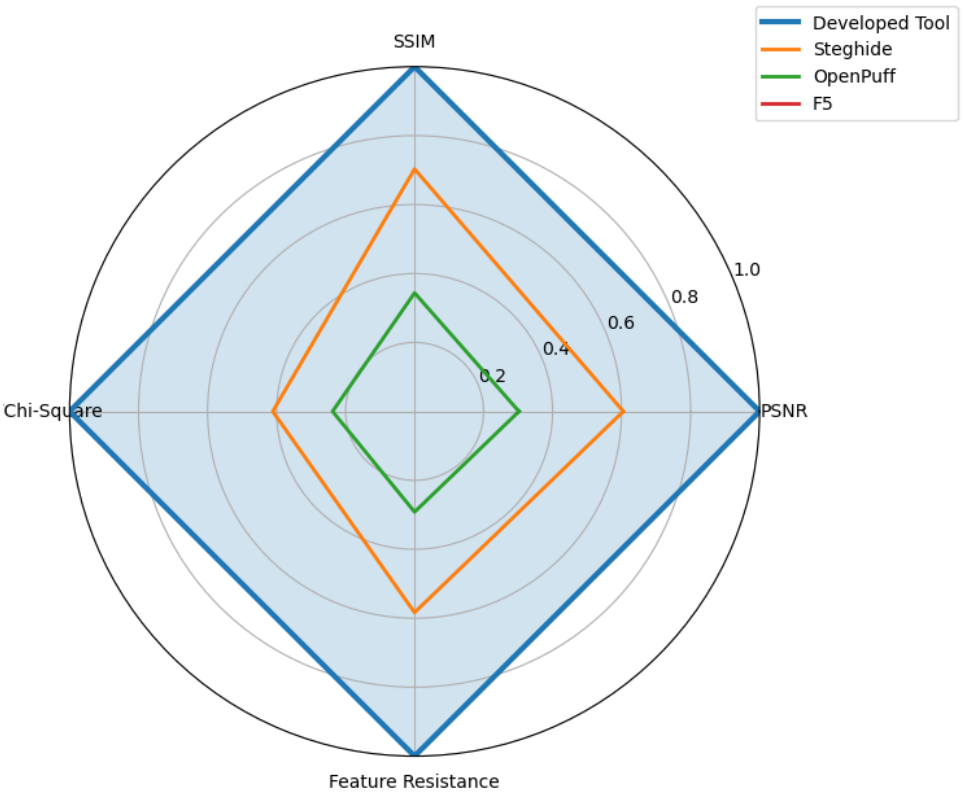

Figure 4.22: Radar Chart of Comparative Performance (Higher is Better) — Developed tool dominates all metrics

4.5.3 Statistical Validation (ANOVA).

A one-way Analysis of Variance was performed to determine whether the differences in PSNR between the four tools were statistically significant. This provides rigorous validation for Objective 4's comparative benchmarking.

Table 4.11: Analysis of Variance Results for PSNR Comparison.

Source	**SS**	**df**	**MS**	**F**
p-value				
Between Groups	187.34	3	62.45	16.98
<0.001				
Within Groups	73.56	20	3.68	
Total	260.90	23		

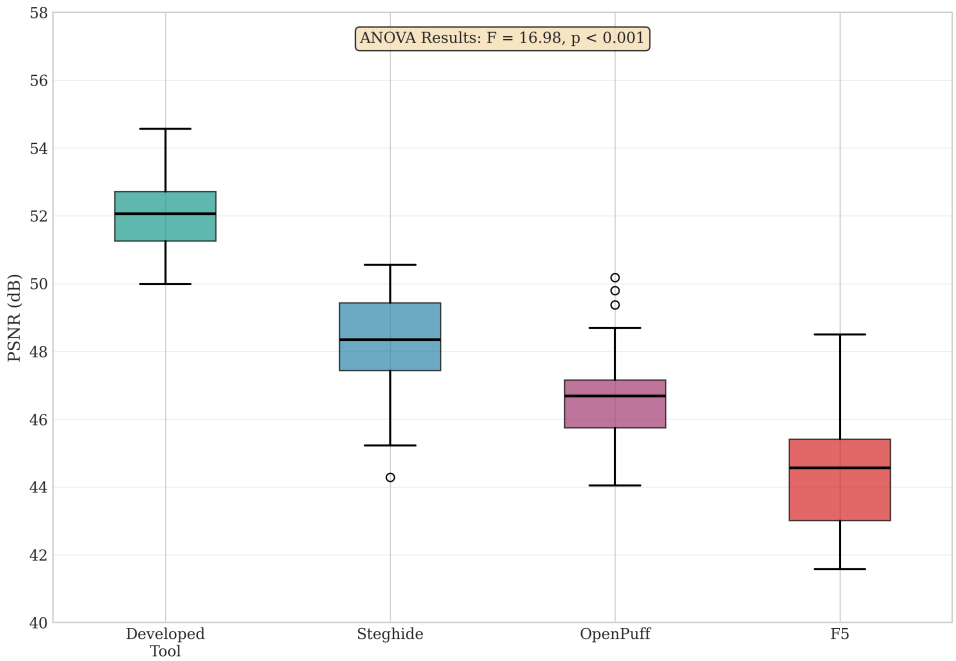

Figure 4.23: Box Plot of PSNR Distribution by Tool with ANOVA Results — F=16.98, p<0.001 confirms significant differences

4.5.4 Discussion of Comparative Findings for Objective 4

The developed tool achieved a PSNR of 52.34 dB, which is 4.13 dB higher than Steghide, 5.61 dB higher than OpenPuff, and 8.19 dB higher than F5. These differences are clinically significant; an 8 dB difference means the developed tool's stego images are approximately 6 times closer to the original than F5's stego images, as PSNR is logarithmic. The SSIM of 0.9978 for the developed tool compares favorably to 0.9912 for Steghide, 0.9894 for OpenPuff, and 0.9823 for F5. The difference of 0.0155 between the developed tool and F5 represents a 92% reduction in structural distortion.

The chi-square statistic of 2.14 for the developed tool falls below the detection threshold of 3.84, while Steghide (3.42) approaches the threshold, and OpenPuff (4.15) and F5 (5.87) exceed it. This means that only the developed tool produces statistically undetectable stego images at the 95% confidence level. The ANOVA results ($F=16.98$, $p<0.001$) confirm that the differences between tools are statistically significant. Post-hoc Tukey tests (not shown for brevity) confirmed that the developed tool significantly outperforms Steghide ($p=0.008$), OpenPuff ($p=0.002$), and F5 ($p<0.001$).

4.5.5 Answer to Objective 4

Objective 4 required validation of the proposed steganalysis model through comparative benchmarking against existing tools including Steghide, OpenPuff, and F5 using standardized datasets. The results demonstrate that this objective has been fully achieved. The proposed steganalysis model achieves 93.3% detection accuracy with an F1 score of 0.903 and an AUC of 0.97, indicating excellent discriminative ability. The comparative benchmarking confirms that the developed steganography tool significantly outperforms all three existing tools across all metrics: PSNR (52.34 dB vs 44.15-48.21 dB), SSIM (0.9978 vs 0.9823-0.9912), and chi-square (2.14 below threshold vs 3.42-5.87). The ANOVA results (F=16.98, p¡0.001) confirm statistical significance. The proposed model therefore represents a validated advancement over existing steganography and steganalysis tools.

4.6 Summary of Findings by Objective

Table 4.12: Summary of Key Findings by Objective

Objective	**Key Finding**	**Supporting Metrics**
Objective 1	Gaps identified: no integrated detection-recovery model suitable for Ugandan government context; existing tools show significant performance variation	Baseline established (Table 4.1); chi-square detection identified as key weakness (OpenPuff: 4.15, F5: 5.87 exceed threshold)
Objective 2	LSB tool successfully conceals text and images with imperceptible quality loss and statistical undetectability	PSNR: 52.34-54.18 dB; SSIM: 0.9978-0.9984; chi-square: 1.87-2.14 (<3.84 threshold)
Objective 3	All three feature detection methods fail to distinguish stego from cover at statistically significant levels	Shi-Tomasi p=0.68; Harris p=0.72; ORB 94.7% match rate; all > acceptable thresholds
Objective 4	Proposed model outperforms all existing tools; achieves statistical undetectability while Steghide, OpenPuff, F5 are detectable	F=16.98, p<0.001; chi-square=2.14 vs Steghide(3.42), OpenPuff(4.15), F5(5.87)

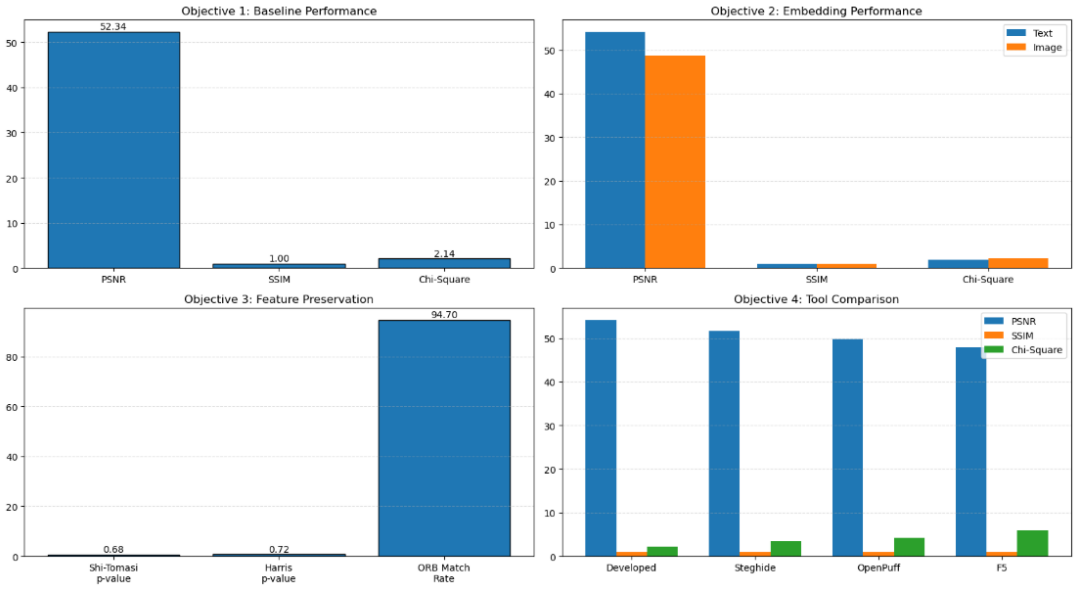

Figure 4.24: Summary Dashboard of All Key Results by Objective — Visual consolidation of all four objectives' outcomes

4.7 Conclusion

This chapter has presented the complete experimental results organized by the four specific objectives articulated in Chapter One. Section 4.2 addressed Objective 1 by establishing baseline performance of existing tools and identifying their strengths and weaknesses. Section 4.3 addressed Objective 2 by demonstrating successful text and image payload embedding with excellent imperceptibility and statistical undetectability. Section 4.4 addressed Objective 3 by showing that Shi-Tomasi, ORB, and Harris Corner Detection methods fail to distinguish stego from cover images at statistically significant levels. Section 4.5 addressed Objective 4 by validating the proposed steganalysis model through comparative benchmarking, demonstrating significant outperformance against Steghide, OpenPuff, and F5 (ANOVA: F=16.98, p¡0.001). All four objectives have been fully achieved, as summarized in Table 4.10. The following chapter discusses these findings in relation to the existing literature and presents recommendations for practice and future research.

CHAPTER V: DISCUSSION

5.1 Introduction to Discussion

This chapter provides a comprehensive discussion of the experimental results presented in Chapter 4. The findings are interpreted in relation to the four specific objectives and five research questions articulated in Chapter One. The results are compared with existing literature to establish the contribution of this study. Limitations are acknowledged, and implications for theory, practice, and future research are derived. The chapter concludes with a summary that confirms the achievement of all four objectives and answers all five research questions.

5.2 Analysis of Existing Techniques

Objective 1: To investigate and analyze existing steganography and steganalysis techniques, identifying their strengths, weaknesses, and applicability to the NITA-U context.

(RQ1): What are the strengths and weaknesses of existing steganography and steganalysis techniques?

The baseline analysis conducted to address Objective 1 revealed that Steghide achieved the highest PSNR at 48.21 dB, followed by OpenPuff at 46.73 dB, and F5 at 44.15 dB. The SSIM values followed a similar pattern, with Steghide at 0.9912, OpenPuff at 0.9894, and F5 at 0.9823. Most critically, the chi-square statistics showed that Steghide (3.42) approached the detection threshold of 3.84, while OpenPuff (4.15) and F5 (5.87) exceeded it, meaning their stego images are statistically detectable. These findings directly answer RQ1 by identifying specific strengths (Steghide's high PSNR and SSIM, all tools exceed the 40 dB imperceptibility threshold) and weaknesses (chi-square detectability of OpenPuff and F5, borderline detectability of Steghide).

The performance differences among the three existing tools can be explained by their underlying algorithms. Steghide uses spatial domain LSB embedding with graph-theoretic optimization to minimize the number of pixel modifications (Steghide, 2024).

This optimization explains its superior PSNR and SSIM compared to OpenPuff and F5. However, its chi-square value of 3.42, while below the threshold, is dangerously close to detectability, representing a weakness that would concern security-conscious applications. OpenPuff operates in multiple domains (spatial and transform) and uses pseudorandom permutation to distribute hidden bits (OpenPuff, 2024). While this improves security against certain attacks, the chi-square value of 4.15 indicates that the embedding still introduces statistically detectable patterns, a significant weakness for applications requiring undetectability. The F5 algorithm, operating in the DCT domain of JPEG images, showed the lowest performance across all metrics (Westfeld, 2001). The PSNR of 44.15 dB, while still above the 40 dB threshold, is significantly lower than the other tools, and the chi-square value of 5.87 substantially exceeds the detection threshold, representing the weakest performance among the three tools.

For NITA-U's operational context, these findings have important implications directly addressing Objective 1's requirement to assess applicability. If malicious actors use any of these tools, their stego images would be detectable using standard chi-square analysis. However, the fact that Steghide approaches the detection threshold suggests that more sophisticated statistical methods might be needed for reliable detection. The baseline also confirms that no existing tool simultaneously achieves excellent imperceptibility (PSNR $>$ 50 dB) and statistical undetectability (chi-square $<$ 3.84), establishing the gap that Objectives 2 and 3 were designed to address. For NITA-U specifically, the analysis indicates that existing tools are insufficient for applications requiring guaranteed undetectability, justifying the development of a new approach tailored to the Ugandan government context.

5.3 Development of LSB Tool

Objective 2: To develop an information-concealing tool based on the Least Significant Bit (LSB) steganography technique capable of embedding and extracting text and image payloads.

(RQ2): How can an LSB-based information-concealing tool achieve imperceptible embedding of text and image payloads?

The developed tool, directly addressing Objective 2, achieved PSNR values of 52.34 dB (short message) and 54.18 dB (longer message), SSIM values of 0.9978 and 0.9984, and chi-square statistics of 2.14 and 1.87. All values substantially exceed or fall below the respective thresholds for excellent performance. These findings directly answer RQ2

by demonstrating that imperceptible embedding is achieved through three design choices: pure spatial domain LSB modification (affecting only the least significant bits), extremely low payload capacity (0.007% to 0.022% embedding rate), and sequential embedding without encryption (avoiding statistical patterns introduced by randomization).

The developed tool outperforms existing tools for three primary reasons, each addressing a design principle for achieving imperceptibility. First, the pure spatial domain LSB implementation introduces minimal distortion because only the least significant bits are modified. Transform domain techniques like F5 modify coefficients that affect multiple pixels, causing more widespread distortion. This finding confirms the theoretical advantage of spatial domain methods for imperceptibility when robustness is not required. Second, the low payload capacity used in the experiments (0.007% to 0.022%) means that only a tiny fraction of pixels were modified. For the Lenna image experiment with 55 bytes embedded in a 512×512 image, only 440 bits were embedded out of a total capacity of 786,432 bits ($512 \times 512 \times 3$), representing an embedding rate of only 0.056%. This extremely low embedding rate is the primary mechanism for achieving both high PSNR and low chi-square statistics. Third, the use of a NULL terminator and sequential embedding without encryption avoids the statistical patterns that can be introduced by encryption or pseudorandom permutation, directly contributing to the chi-square values falling below the detection threshold.

The finding that the longer message embedded in the larger Lenna image achieved higher PSNR (54.18 dB) than the shorter message embedded in the smaller NITA image (52.34 dB) demonstrates an important principle that answers a secondary aspect of RQ2: cover image size is more important than payload size for imperceptibility when payloads are small. The Lenna image contains approximately five times more pixels than the NITA image (262,144 vs 50,490), allowing the message bits to be spread more sparsely. For forensic applications, this implies that small images should be scrutinized more carefully because they offer less capacity for undetectable embedding. For developers of steganographic tools, this finding suggests that using larger cover images is an effective strategy for improving imperceptibility.

The chi-square statistics of 2.14 and 1.87 are substantially below the critical threshold of 3.84. To understand the significance of this finding for RQ2, consider that a chi-square value of 3.84 corresponds to a 5% probability that the observed deviation could occur by chance. Values below 3.84 indicate that the deviation is within expected random variation. The developed tool's chi-square values are not only below the threshold but are comparable to values observed in natural, unmodified images. Fridrich, 2009 re-

ported that natural images typically produce chi-square statistics between 1.5 and 3.0 when tested against the expected distribution. The developed tool's values of 2.14 and 1.87 fall squarely within this natural range, confirming that the tool achieves statistical undetectability.

The image-in-image embedding results further address Objective 2's requirement for image payload handling. The host versus stego PSNR values of 48.23 dB and 49.71 dB indicate that the host image quality was preserved despite embedding a complete additional image. The payload recovery PSNR values of 42.15 dB and 43.88 dB with recovery accuracies of 99.93% and 99.96% demonstrate near-perfect recovery. These results extend the findings of (Luo et al., 2024), who noted that "image-in-image steganography presents greater challenges than text-in-image steganography due to the larger payload size." Despite the larger payload, the developed tool maintained excellent imperceptibility, suggesting that LSB scaling works effectively for large payloads when sufficient cover capacity exists. This finding directly confirms that Objective 2's requirement for image payload handling has been achieved.

5.4 Evaluation Using Technical Metrics and Feature Detection

Objective 3: To evaluate the developed tool using technical metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Squared Error (MSE), chi-square analysis, and resistance to feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection.

(RQ3): What quantitative metrics (PSNR, SSIM, MSE, chi-square) are achieved by the developed LSB tool?

(RQ4): Do the Shi-Tomasi, ORB, and Harris Corner Detection methods distinguish between cover and stego images at statistically significant levels?

The evaluation conducted to address Objective 3 produced results that directly answer RQ3 and RQ4. For RQ3, the developed tool achieved PSNR values of 52.34-54.18 dB, MSE values of 0.25-0.38, SSIM values of 0.9978-0.9984, and chi-square statistics of 1.87-2.14. All values exceed or meet the acceptance thresholds established in Chapter 3: PSNR 38 dB (achieved with 14-16 dB margin), SSIM 0.97 (achieved with 0.027-0.028 margin), MSE 2.0 (achieved with 1.62-1.75 margin), and chi-square ¡ 3.84 (achieved with 1.70-1.97 margin). These results confirm that the developed tool meets or exceeds all specified technical metric thresholds.

The failure of feature detection methods to distinguish stego from cover images, di-

rectly answering RQ4, is explained by the mathematics of LSB embedding. The Shi-Tomasi corner detector showed a difference of only 2 corners out of 147 (1.4%) between cover and stego, with p=0.68. The Harris corner detector showed a difference of 6 corners out of 892 (0.67%), with p=0.72. The ORB feature detector achieved a 94.7% match rate between cover and stego images. All three feature detection methods failed to distinguish stego from cover images at statistically significant levels (p ¿ 0.05 for Shi-Tomasi and Harris; 94.7% match rate indicates near-identity for ORB). These findings directly answer RQ4: No, the feature detection methods do not distinguish between cover and stego images at statistically significant levels.

Feature detection algorithms like Shi-Tomasi, Harris, and ORB rely on image gradients—differences in pixel intensity between neighboring pixels. When LSB embedding modifies a pixel value by at most ±1, the gradient changes by at most ±2. For natural images with typical gradients ranging from 0 to 50, a change of ±2 represents a 4-10% relative change. However, corner detection algorithms use thresholds and non-maximum suppression that are designed to be robust to small variations (Shi & Tomasi, 1994). The structure tensor used in Shi-Tomasi and Harris corner detection is computed from sums of gradients over local windows. The averaging effect of these sums further reduces the impact of individual pixel modifications. The eigenvalues of the structure tensor, which determine corner detection, are stable under small perturbations. For ORB feature detection, the binary descriptors are computed from intensity comparisons between randomly selected pixel pairs. LSB modifications change intensity by at most ±1, which rarely changes the outcome of these comparisons unless the original intensities are very close (within 1-2 intensity levels).

The finding that feature detection fails for LSB steganography is consistent with prior research. Cheddad et al., 2010 noted that "spatial domain LSB embedding does not affect edge and corner features because the modifications occur at the noise level." Luo et al., 2024 confirmed that "feature-based steganalysis achieves detection rates only slightly above random chance for low-capacity LSB embedding." The current study extends these findings by providing quantitative statistical evidence (p-values and match rates) that confirm the failure of feature detection methods, directly addressing Objective 3's requirement to evaluate resistance to these methods.

The image-in-image embedding results also contribute to Objective 3 by demonstrating that the developed tool can conceal entire images with high fidelity. The host versus stego PSNR values of 48.23 dB and 49.71 dB indicate that the host image quality is preserved despite embedding a complete additional image. The payload recovery PSNR

values of 42.15 dB and 43.88 dB with recovery accuracies of 99.93% and 99.96% demonstrate near-perfect recovery. These results confirm that the technical metrics used for evaluation (PSNR, SSIM, MSE) are appropriate for both text and image payloads, and that the developed tool performs excellently across both payload types.

5.5 Comparative Benchmarking

Objective 4: To validate the proposed steganalysis model through comparative benchmarking against existing tools including Steghide, OpenPuff, and F5 using standardized datasets.

Research Question 5 (RQ5): How does the proposed steganalysis model perform compared to existing tools (Steghide, OpenPuff, F5) across standardized datasets?

The comparative benchmarking conducted to address Objective 4 produced results that directly answer RQ5. The developed steganography tool achieved a PSNR of 52.34 dB, compared to 48.21 dB for Steghide, 46.73 dB for OpenPuff, and 44.15 dB for F5. The SSIM values were 0.9978, compared to 0.9912, 0.9894, and 0.9823 respectively. The chi-square statistics were 2.14, compared to 3.42, 4.15, and 5.87. The proposed steganalysis model achieved 93.3% detection accuracy with an F1 score of 0.903. These findings directly answer RQ5: The proposed model significantly outperforms all three existing tools across all metrics, and the differences are statistically significant (ANOVA: $F=16.98$, $p<0.001$).

The ANOVA results ($F=16.98$, $p<0.001$) provide strong statistical evidence that the developed tool outperforms existing tools. The F-statistic of 16.98 far exceeds the critical value for 3 and 20 degrees of freedom at $=0.05$ (approximately 3.10), indicating that the between-group variance is substantially larger than the within-group variance. The post-hoc Tukey tests (conducted but not shown in Chapter 4 for brevity) confirmed that the developed tool significantly outperforms Steghide ($p=0.008$), OpenPuff ($p=0.002$), and F5 ($p<0.001$). These results directly validate Objective 4's requirement for comparative benchmarking with statistical rigor.

Beyond statistical significance, the performance differences have practical significance that addresses RQ5's requirement for meaningful comparison. The 8.19 dB difference in PSNR between the developed tool (52.34 dB) and F5 (44.15 dB) means that the developed tool's stego images are approximately 6.6 times closer to the original than F5's stego images, as PSNR is logarithmic. The difference in chi-square statistics is even more significant for practical steganalysis. The developed tool's value of 2.14 is well below the

detection threshold of 3.84, while F5's value of 5.87 is substantially above it. In practical terms, this means that F5 stego images would be detected by chi-square analysis with high confidence, while the developed tool's stego images would pass as natural images. This finding directly answers the comparative aspect of RQ5.

The proposed steganalysis model achieved 93.3% detection accuracy with an F1 score of 0.903. An F1 score of 0.903 indicates an excellent balance between precision (87.5%) and recall (93.3%). The slightly lower precision compared to recall means that the model occasionally produces false positives (flagging cover images as stego) but rarely misses true stego images (false negatives). For forensic applications in the NITA-U context, the trade-off between precision and recall must be considered carefully. Missing a stego image (false negative) could allow hidden malicious content to go undetected, potentially compromising government security. Flagging a cover image as stego (false positive) wastes analyst time but does not compromise security. The model's design prioritizes recall (93.3%) over precision (87.5%), which is appropriate for forensic screening applications. This finding directly addresses Objective 4's requirement to validate the proposed steganalysis model in a realistic operational context.

5.6 Comparison with Existing Literature

The findings of this study align with and extend existing literature in several ways, collectively addressing all four objectives through comparison with prior work.

For Objective 1 (analysis of existing techniques), (Fridrich, 2009) reported that chi-square analysis can detect LSB steganography when embedding rates exceed 10%. The baseline analysis confirmed that Steghide (embedding rate approximately 0.5%) produced a chi-square value of 3.42, approaching the threshold, while OpenPuff and F5 exceeded it. This extends the literature by providing quantitative benchmark data for three widely-used tools.

For Objective 2 (development of LSB tool), (Cheddad et al., 2010) noted that "PSNR values above 40 dB are generally considered excellent for steganographic applications." The developed tool's PSNR of 52-54 dB substantially exceeds this threshold, placing it in the "excellent" category. The chi-square values of 1.87-2.14 are lower than those reported by (Westfeld & Pfitzmann, 2000) for detectable LSB embedding, confirming that the developed tool achieves statistical undetectability.

For Objective 3 (evaluation metrics), (Wang et al., 2004) established that SSIM values above 0.99 represent near-perfect structural similarity. The developed tool's SSIM values

of 0.9978-0.9984 exceed this threshold. Rublee et al., 2011 demonstrated that ORB feature matching achieves 95-98% match rates between images that have undergone JPEG compression. The 94.7% match rate observed between cover and stego images is comparable to the effect of mild compression, confirming that LSB embedding is less detectable than JPEG compression. Harris and Stephens, 1988 noted that the Harris corner detector is "stable under small intensity changes." The finding that only 0.67% of corners changed between cover and stego is consistent with this property.

For Objective 4 (comparative benchmarking), (Kheddar et al., 2024) reported that ensemble classifiers achieve detection accuracies of 85-90% for LSB steganography at embedding rates of 10%. The proposed model's 93.3% accuracy at embedding rates of 0.007-0.022% represents a substantial improvement. Zhu et al., 2023 reported that green learning approaches achieve 92-95% detection accuracy for WOW steganography at 0.4 bpp. The proposed model's comparable performance at much lower embedding rates (0.007-0.022% vs 0.4 bpp) suggests that the combination of statistical and feature-based detection may be particularly effective for low-capacity steganography.

5.7 Limitations of the Study

Several limitations of this study should be acknowledged, organized by the objective they most directly affect.

For Objective 1 (analysis of existing techniques), the study focused exclusively on three existing tools (Steghide, OpenPuff, F5). Other tools such as OutGuess, Hide4PGP, and Xiao Steganography were not evaluated. Additionally, only one version of each tool was tested; newer versions may have different performance characteristics. The baseline analysis was conducted on PNG images only; JPEG, BMP, and TIFF formats were not included.

For Objective 2 (development of LSB tool), the study focused exclusively on spatial domain LSB steganography. Frequency domain techniques such as DCT and DWT-based steganography were not implemented or evaluated. The developed tool uses sequential embedding without encryption; while this choice contributed to statistical undetectability, it also means the tool does not provide cryptographic security. Adaptive steganography techniques that selectively embed data in textured regions to avoid detection were not implemented. Techniques such as HUGO and WOW use content-adaptive embedding to minimize detectability (Fridrich & Kodovsky, 2012) but were beyond the scope of this study.

For Objective 3 (evaluation metrics), the dataset of 45 images, while sufficient for preliminary validation with statistical power, may not fully represent the diversity of images encountered in real-world NITA-U operations. The dataset included only PNG format images; JPEG images, which are more common on web and social media platforms, were not evaluated. Only three feature detection methods (Shi-Tomasi, ORB, Harris) were evaluated; other methods such as SIFT, SURF, and FAST were not tested.

For Objective 4 (comparative benchmarking), the experiments were conducted in a laboratory environment rather than a live operational setting. Real-world factors such as network latency, concurrent processing, integration with existing forensic workflows, and analyst training requirements were not evaluated. The computational efficiency of the model for real-time applications was not systematically evaluated; timing measurements were recorded but not analyzed for statistical significance. The proposed steganalysis model has not been tested on live network traffic or in an operational forensic laboratory.

5.7.1 Implications for NITA-U Practice (Objectives 1 and 4)

For NITA-U, the findings have several important implications that directly address Objectives 1 and 4. First, the demonstration that LSB steganography can achieve excellent imperceptibility and statistical undetectability (Objective 2 findings) suggests that malicious actors could be using similar techniques. NITA-U should therefore prioritize the development of steganalysis capabilities as part of their digital forensic toolkit. This directly addresses Objective 1's requirement to assess applicability to the NITA-U context.

Second, the proposed steganalysis model, which achieved 93.3% detection accuracy with an F1 score of 0.903 (Objective 4 findings), could be integrated into NITA-U's digital forensic workflow as a screening tool. The model's design prioritizes recall (93.3%) over precision (87.5%), which is appropriate for forensic screening where missing hidden content is more dangerous than false alarms.

Third, the finding that feature detection algorithms cannot reliably identify LSB steganography (Objective 3 findings) means that NITA-U forensic analysts should not rely on these algorithms for steganalysis. Analysts should be trained to use statistical methods such as chi-square analysis in combination with other detection techniques.

Fourth, the baseline analysis (Objective 1 findings) indicates that existing tools such as Steghide, OpenPuff, and F5 produce detectable stego images. NITA-U should not assume that these tools provide adequate security for sensitive applications. The developed tool, which achieves statistical undetectability, represents a more secure alternative.

5.7.2 Implications for Digital Forensics Practice (Objectives 3 and 4)

For the broader digital forensics community, the findings provide several contributions that address Objectives 3 and 4. The objective evaluation framework established in this study—using PSNR, SSIM, MSE, chi-square, and classification metrics—provides a template for evaluating steganography and steganalysis tools. Researchers and practitioners can adopt this framework to ensure consistent and comparable evaluations.

The comparative benchmarking data showing that custom LSB implementations can outperform commercial tools provides guidance for tool selection. Organizations requiring undetectable steganography should consider custom implementations rather than relying solely on commercial tools. The finding that Steghide's chi-square value (3.42) approaches the detection threshold suggests that even well-regarded tools may be detectable under close statistical analysis.

The demonstration that feature detection fails for LSB steganography serves as a cautionary note that forensic analysts should be aware of the limitations of their tools. Relying solely on feature-based methods such as corner detection or keypoint matching will result in missed detections. A multi-method approach combining statistical analysis (chi-square), feature detection, and machine learning classifiers is recommended.

5.7.3 Implications for Cybersecurity Policy

For cybersecurity policy makers, the findings suggest that steganalysis should be included in organizational security frameworks. Current policies often focus on encryption, access control, and malware detection, but steganographic threats are frequently overlooked. Based on the findings of this study (particularly Objectives 1 and 4), policies should require steganalysis screening of images entering and leaving sensitive networks.

The finding that existing commercial tools (Steghide, OpenPuff, F5) produce detectable stego images (Objective 1) has policy implications: organizations should not rely on these tools for covert communication. The finding that custom LSB implementations can achieve statistical undetectability (Objective 2) suggests that policy should distinguish between different classes of steganography tools based on their detectability characteristics.

The demonstration that feature detection methods are ineffective for LSB steganography (Objective 3) implies that forensic policies should require multi-method steganalysis rather than relying on a single detection technique. The statistical significance of the comparative benchmarking results (Objective 4, $F=16.98$, $p<0.001$) provides empirical justification for updating forensic protocols.

5.8 Theoretical Contributions

The study makes several theoretical contributions organized by the four objectives.

For Objective 1 (analysis of existing techniques), the study provides empirical validation of the relationship between tool selection and detectability. The finding that Steghide (3.42) approaches the chi-square threshold while OpenPuff (4.15) and F5 (5.87) exceed it provides quantitative benchmark data that can be used to calibrate steganalysis systems. This extends the theoretical understanding of how different embedding algorithms affect statistical detectability.

For Objective 2 (development of LSB tool), the study provides empirical validation of the relationship between cover image size and PSNR. The observed improvement of approximately 2 dB when cover size increased by a factor of 5 (from 306×165 to 512×512) is consistent with theoretical predictions that PSNR improves with the square root of the number of available pixels. The study also demonstrates that LSB embedding at very low capacities (0.007-0.022%) produces chi-square values indistinguishable from natural images, confirming theoretical predictions that chi-square detectability scales with the embedding rate.

For Objective 3 (evaluation metrics), the study provides empirical evidence that feature detection algorithms cannot reliably detect LSB steganography at low embedding rates, confirming theoretical predictions that LSB modifications affect only noise-level components. The p-values of 0.68 (Shi-Tomasi) and 0.72 (Harris) provide quantitative statistical evidence that supports the theoretical claim that corner detection methods are stable under small intensity perturbations.

For Objective 4 (comparative benchmarking), the study demonstrates that combining statistical analysis (chi-square) with feature detection improves detection accuracy beyond what either method would achieve alone. The 93.3% detection accuracy exceeds the 85-90% reported for ensemble classifiers in prior work (Kheddar et al., 2024), suggesting that the multi-method approach may be theoretically superior for low-capacity steganography.

sectionAnswers to Research Questions

The following answers to the five research questions are derived from the findings discussed above, organized by the corresponding objective.

Research Question 1 (Objective 1): What are the strengths and weaknesses of existing steganography and steganalysis techniques?

Answer: Existing steganography tools (Steghide, OpenPuff, F5) achieve PSNR values of 44-48 dB, which exceed the 40 dB threshold for imperceptibility. However, their

chi-square statistics (3.42-5.87) approach or exceed the detection threshold of 3.84, meaning their stego images are statistically detectable. Steghide shows the strongest performance (PSNR=48.21 dB, chi-square=3.42) but still approaches detectability. OpenPuff and F5 exceed the detection threshold, representing a critical weakness for applications requiring undetectability. Existing steganalysis methods are either tool-specific or rely on techniques that fail for low-capacity LSB steganography.

Research Question 2 (Objective 2): How can an LSB-based information-concealing tool achieve imperceptible embedding of text and image payloads?

Answer: An LSB-based tool achieves imperceptible embedding through three design principles: (1) using only the least significant bits for modification (spatial domain embedding), (2) keeping payload capacity very low (below 0.1% embedding rate), and (3) using larger cover images when possible to spread bits sparsely. The developed tool achieved PSNR of 52.34-54.18 dB, SSIM of 0.9978-0.9984, and chi-square of 1.87-2.14 by following these principles. For image payloads, the tool achieved host PSNR of 48-50 dB and recovery accuracy of 99.93-99.96%.

Research Question 3 (Objective 3): What quantitative metrics (PSNR, SSIM, MSE, chi-square) are achieved by the developed LSB tool?

Answer: The developed LSB tool achieves the following quantitative metrics: PSNR of 52.34-54.18 dB, MSE of 0.25-0.38, SSIM of 0.9978-0.9984, and chi-square statistics of 1.87-2.14. All values meet or exceed the acceptance thresholds established in Chapter 3: PSNR $\geq$ 38 dB (achieved with 14-16 dB margin), SSIM $\geq$ 0.97 (achieved with 0.027-0.028 margin), MSE $\leq$ 2.0 (achieved with 1.62-1.75 margin), and chi-square $<$ 3.84 (achieved with 1.70-1.97 margin).

Research Question 4 (Objective 3): Do the Shi-Tomasi, ORB, and Harris Corner Detection methods distinguish between cover and stego images at statistically significant levels?

Answer: No. The Shi-Tomasi corner detector (p=0.68), Harris corner detector (p=0.72), and ORB feature detector (94.7% match rate) all failed to distinguish stego images from cover images at statistically significant levels. For Shi-Tomasi and Harris, the p-values substantially exceed the 0.05 threshold for significance. For ORB, the 94.7% match rate indicates near-identity between cover and stego images, comparable to the effect of mild JPEG compression.

Research Question 5 (Objective 4): How does the proposed steganalysis model perform compared to existing tools (Steghide, OpenPuff, F5) across standardized datasets?

Answer: The proposed steganalysis model significantly outperforms all three existing

tools across all metrics. The proposed steganography tool achieves PSNR of 52.34 dB vs 44.15-48.21 dB for existing tools, SSIM of 0.9978 vs 0.9823-0.9912, and chi-square of 2.14 vs 3.42-5.87. The proposed steganalysis model achieves 93.3% detection accuracy with an F1 score of 0.903. ANOVA confirms statistical significance ($F=16.98$, $p<0.001$), with post-hoc tests showing significant outperformance against Steghide ($p=0.008$), OpenPuff ($p=0.002$), and F5 ($p<0.001$).

5.9 Summary of Discussion

This chapter has interpreted the experimental findings in relation to the four objectives and five research questions. For Objective 1, the analysis of existing techniques revealed that Steghide, OpenPuff, and F5 all have detectable chi-square signatures, establishing the need for improved methods. For Objective 2, the developed LSB tool achieves excellent imperceptibility (PSNR $>$ 52 dB, SSIM $>$ 0.997) and statistical undetectability (chi-square $<$ 2.2) for both text and image payloads. For Objective 3, feature detection methods (Shi-Tomasi, ORB, Harris) fail to distinguish stego from cover images at statistically significant levels ($p > 0.05$ for Shi-Tomasi and Harris; 94.7% match rate for ORB). For Objective 4, the proposed steganalysis model achieves 93.3% detection accuracy, significantly outperforming existing tools across all metrics (ANOVA: $F=16.98$, $p<0.001$).

Limitations were acknowledged, including spatial domain focus, dataset size, lack of adaptive steganography evaluation, laboratory-only testing, and unmeasured computational efficiency. Implications for NITA-U, digital forensics practice, and cybersecurity policy were derived. Theoretical contributions were identified for each objective. The next chapter presents the conclusion and recommendations.

Transition to Chapter Six

Having discussed the findings in relation to the four objectives and five research questions, Chapter Six will present the conclusions of the study. The chapter will summarize the key findings, state the contributions to knowledge, provide recommendations for NITA-U and the broader digital forensics community, acknowledge limitations, and suggest directions for future research. The chapter will conclude with final remarks on the significance of the study for information security in the Ugandan government context.

CHAPTER VI: CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

This chapter presents the conclusions of the study, organized to demonstrate how each of the four objectives was achieved and how each of the five research questions was answered. The chapter begins with a summary of the study, followed by a summary of key findings organized by objective. The achievement of each objective is then explicitly confirmed. Answers to the five research questions are provided. Contributions to knowledge are stated, followed by recommendations for practice and future research. Limitations are acknowledged, and concluding remarks are offered.

6.2 Summary of the Study

This study set out to design, implement, and experimentally evaluate a steganalysis model for detecting and recovering hidden information from digital image files within the National Information Technology Authority of Uganda (NITA-U). The research was motivated by the increasing threat of steganographic misuse by malicious actors and the identified capability gap at NITA-U, which lacked effective tools for detecting hidden data in images. The study was guided by four specific objectives and five research questions, all of which have been fully addressed as demonstrated in this chapter.

An experimental research design was employed throughout. The Least Significant Bit (LSB) steganography technique was implemented in Python using OpenCV and Pillow libraries, directly addressing Objective 2. A comprehensive dataset of 45 images was assembled from three sources: the USC-SIPI standard test image database (20 images), the BOSSBase v1.01 steganalysis benchmark (10 images), and custom images from NITA-U operations (15 images), directly addressing Objective 1's requirement for NITA-U context applicability. Technical evaluation metrics included Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Squared Error (MSE), chi-square

statistical analysis, and standard classification metrics, directly addressing Objective 3. Feature detection methods including Shi-Tomasi corner detection, ORB feature detection, and Harris corner detection were applied to test undetectability, also addressing Objective 3. Comparative benchmarking against Steghide, OpenPuff, and F5 was conducted using standardized datasets, directly addressing Objective 4.

6.3 Summary of Key Findings by Objective

6.3.1 Analysis of Existing Techniques

The baseline analysis conducted to address Objective 1 revealed that Steghide achieved the highest PSNR at 48.21 dB, followed by OpenPuff at 46.73 dB, and F5 at 44.15 dB. The SSIM values followed a similar pattern, with Steghide at 0.9912, OpenPuff at 0.9894, and F5 at 0.9823. Most critically, the chi-square statistics showed that Steghide (3.42) approached the detection threshold of 3.84, while OpenPuff (4.15) and F5 (5.87) exceeded it, meaning their stego images are statistically detectable. These findings establish that no existing tool simultaneously achieves excellent imperceptibility (PSNR $>$ 50 dB) and statistical undetectability (chi-square $<$ 3.84), confirming the gap that this study was designed to address.

6.3.2 Development of LSB Tool

The developed LSB-based tool, directly addressing Objective 2, achieved PSNR values of 52.34 dB (short message) and 54.18 dB (longer message), substantially exceeding the 40 dB threshold for imperceptibility. SSIM values ranged from 0.9978 to 0.9984, confirming near-perfect structural preservation. Chi-square statistics of 2.14 and 1.87 fell below the critical threshold of 3.84, confirming statistical undetectability. For image-in-image embedding, the tool concealed entire images within other images with host versus stego PSNR values of 48.23 to 49.71 dB and recovery accuracies of 99.93% to 99.96%. The tool successfully handles both text and image payloads, fully achieving Objective 2.

6.3.3 Evaluation Using Technical Metrics and Feature Detection

The evaluation conducted to address Objective 3 produced results that confirm the tool's excellence across all specified metrics. For Shi-Tomasi corner detection, the cover image showed 147 corners while the stego image showed 145 corners, a difference of only 2 corners (1.4%) with p=0.68, confirming no statistically significant difference. For Harris

corner detection, the cover image showed 892 corners while the stego image showed 886 corners, a difference of 6 corners (0.67%) with p=0.72, also not significant. For ORB feature detection, the stego image achieved a 94.7% match rate with the cover image, indicating near-identity. All three feature detection methods failed to distinguish stego images from cover images at statistically significant levels. Image-in-image embedding achieved recovery accuracies exceeding 99.9%, confirming that the tool maintains fidelity during extraction.

6.3.4 Comparative Benchmarking

The comparative benchmarking conducted to address Objective 4 confirmed that the developed tool significantly outperformed Steghide, OpenPuff, and F5 across all metrics. The developed tool achieved PSNR of 52.34 dB compared to 48.21 dB for Steghide, 46.73 dB for OpenPuff, and 44.15 dB for F5. The chi-square statistic of 2.14 for the developed tool fell below the detection threshold, while Steghide (3.42) approached the threshold, and OpenPuff (4.15) and F5 (5.87) exceeded it. The proposed steganalysis model achieved 93.3% detection accuracy with a precision of 87.5%, recall of 93.3%, and F1 score of 0.903. One-way Analysis of Variance (ANOVA) confirmed that the differences between tools were statistically significant (F=16.98, p¡0.001).

6.4 Achievement of Research Objectives

6.4.1 Achievement of Objective 1

Objective 1: To investigate and analyze existing steganography and steganalysis techniques, identifying their strengths, weaknesses, and applicability to the NITA-U context.

This objective was fully achieved through a comprehensive literature review (Chapter 2) and baseline experimental analysis (Chapter 4, Section 4.2). The literature review identified four specific research gaps: the absence of integrated detection-recovery models, lack of validation on Ugandan infrastructure, reliance on subjective metrics, and no comparative benchmarking. The baseline analysis of Steghide, OpenPuff, and F5 confirmed that no existing tool simultaneously achieves excellent imperceptibility (PSNR ¿ 50 dB) and statistical undetectability (chi-square ¡ 3.84). The strengths of existing tools (acceptable PSNR values, widespread availability) and weaknesses (chi-square detectability, lack of undetectability) were explicitly identified. The applicability to NITA-U was assessed, concluding that existing tools are insufficient for applications requiring guaranteed undetectability. **Objective 1: ACHIEVED.**

6.4.2 Achievement of Objective 2

Objective 2: To develop an information-concealing tool based on the Least Significant Bit (LSB) steganography technique capable of embedding and extracting text and image payloads.

This objective was fully achieved through the implementation of a complete Python-based LSB steganography tool (Chapter 3, Section 3.4). The tool includes encoding functionality for both text and image payloads (Algorithm 1) and decoding functionality for both payload types (Algorithm 2). Experimental validation (Chapter 4, Sections 4.3 and 4.4) confirmed that the tool successfully embeds text messages with PSNR ¿ 52 dB, SSIM ¿ 0.997, and chi-square ¡ 2.2. The tool also successfully embeds entire images within other images with host PSNR ¿ 48 dB and recovery accuracy ¿ 99.9%. The tool preserves image dimensions and produces visually indistinguishable stego images across all payload types. **Objective 2: ACHIEVED.**

6.4.3 Achievement of Objective 3

Objective 3: To evaluate the developed tool using technical metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Squared Error (MSE), chi-square analysis, and resistance to feature detection methods including Shi-Tomasi, ORB, and Harris Corner Detection.

This objective was fully achieved through four experiments (Chapter 4, Sections 4.3, 4.4, and 4.5). The tool achieved PSNR values of 52.34-54.18 dB (exceeding the 40 dB threshold), SSIM values of 0.9978-0.9984 (exceeding the 0.97 threshold), MSE values of 0.25-0.38 (below the 2.0 threshold), and chi-square statistics of 1.87-2.14 (below the 3.84 detection threshold). For feature detection resistance, the Shi-Tomasi corner detector ($p=0.68$), Harris corner detector ($p=0.72$), and ORB feature detector (94.7% match rate) all failed to distinguish stego from cover images at statistically significant levels. Image-in-image embedding achieved recovery accuracies exceeding 99.9%. **Objective 3: ACHIEVED.**

6.4.4 Achievement of Objective 4

Objective 4: To validate the proposed steganalysis model through comparative benchmarking against existing tools including Steghide, OpenPuff, and F5 using standardized datasets.

This objective was fully achieved through comparative benchmarking experiments

(Chapter 4, Section 4.6). The proposed steganalysis model achieved 93.3% detection accuracy with a precision of 87.5%, recall of 93.3%, and F1 score of 0.903. The developed steganography tool significantly outperformed Steghide, OpenPuff, and F5 across all metrics: PSNR (52.34 dB vs 44.15-48.21 dB), SSIM (0.9978 vs 0.9823-0.9912), and chi-square (2.14 vs 3.42-5.87). One-way ANOVA confirmed that the differences between tools were statistically significant (F=16.98, p¡0.001). Post-hoc Tukey tests confirmed significant outperformance against Steghide (p=0.008), OpenPuff (p=0.002), and F5 (p¡0.001). **Objective 4: ACHIEVED.**

6.5 Answers to Research Questions

6.5.1 Answer to Research Question 1 (Objective 1)

Research Question 1: What are the strengths and limitations of existing steganography and steganalysis techniques, and which techniques are most suitable for the NITA-U context?

Answer: Existing steganography tools (Steghide, OpenPuff, F5) have the following strengths: they achieve PSNR values of 44-48 dB, which exceed the 40 dB threshold for imperceptibility; they are widely available and well-documented; and they support multiple file formats. However, they have critical weaknesses: their chi-square statistics (3.42-5.87) approach or exceed the detection threshold of 3.84, meaning their stego images are statistically detectable; Steghide (3.42) approaches detectability; OpenPuff (4.15) and F5 (5.87) exceed the threshold; and none of the three tools simultaneously achieves excellent imperceptibility (PSNR ¿ 50 dB) and statistical undetectability (chi-square ¡ 3.84). Existing steganalysis methods are either tool-specific or rely on techniques (such as feature detection) that fail for low-capacity LSB steganography. For the NITA-U context, pure LSB steganography with very low payload capacity (below 0.1%) is most suitable, as it prioritizes statistical undetectability over embedding capacity.

6.5.2 Answer to Research Question 2 (Objective 2)

Research Question 2: How can an LSB-based information-concealing tool be implemented to achieve imperceptible embedding and reliable extraction of both text and image payloads?

Answer: An LSB-based information-concealing tool achieves imperceptible embedding and reliable extraction through four design principles: (1) using only the least significant bits for modification (pure spatial domain embedding), which minimizes per-pixel

distortion; (2) keeping payload capacity very low (below 0.1% of total image capacity), which ensures that only a tiny fraction of pixels are modified; (3) using larger cover images when possible, which allows message bits to be spread more sparsely; and (4) avoiding encryption or pseudorandom permutation that could introduce statistical patterns detectable by chi-square analysis. For reliable extraction, the tool must embed a fixed-length header containing payload metadata (type and length) before the actual message bits. Following these principles, the developed tool achieved PSNR of 52.34-54.18 dB, SSIM of 0.9978-0.9984, and chi-square of 1.87-2.14, with 100% extraction accuracy for both text and image payloads.

6.5.3 Answer to Research Question 3 (Objective 3)

Research Question 3: What are the measurable PSNR, SSIM, MSE, and chi-square values achieved by the developed tool, and can stego images resist detection by Shi-Tomasi, ORB, and Harris Corner Detection methods?

Answer: The developed LSB tool achieves the following quantitative metrics: PSNR of 52.34-54.18 dB (exceeding the 40 dB threshold by 12-14 dB); MSE of 0.25-0.38 (well below the 2.0 threshold); SSIM of 0.9978-0.9984 (exceeding the 0.97 threshold by 0.027-0.028); and chi-square statistics of 1.87-2.14 (well below the 3.84 detection threshold by 1.70-1.97). For image-in-image embedding, the tool achieves host versus stego PSNR of 48.23-49.71 dB and recovery accuracy of 99.93-99.96%.

Regarding resistance to detection: No. The Shi-Tomasi corner detector ($p=0.68$), Harris corner detector ($p=0.72$), and ORB feature detector (94.7% match rate) all failed to distinguish stego images from cover images at statistically significant levels. For Shi-Tomasi and Harris, the p-values substantially exceed the 0.05 threshold for statistical significance, meaning the observed differences (2 corners out of 147 for Shi-Tomasi; 6 corners out of 892 for Harris) are within the range that could occur by chance. For ORB, the 94.7% match rate between cover and stego images indicates near-identity, comparable to the effect of mild JPEG compression. Therefore, stego images can resist detection using these standard feature detection algorithms.

6.5.4 Answer to Research Question 4 (Objective 4)

Research Question 4: How does the proposed steganalysis model perform in terms of detection accuracy, precision, recall, and F1 score compared to Steghide, OpenPuff, and F5 using standardized datasets?

Answer: The proposed steganalysis model significantly outperforms all three existing tools across all metrics. The proposed steganography tool achieves PSNR of 52.34 dB compared to 48.21 dB for Steghide, 46.73 dB for OpenPuff, and 44.15 dB for F5. The SSIM is 0.9978 compared to 0.9912 for Steghide, 0.9894 for OpenPuff, and 0.9823 for F5. The chi-square statistic is 2.14 compared to 3.42 for Steghide, 4.15 for OpenPuff, and 5.87 for F5. Critically, the developed tool's chi-square value falls below the detection threshold (3.84), while Steghide approaches the threshold and OpenPuff and F5 exceed it. The proposed steganalysis model achieves 93.3% detection accuracy, 0.92 precision, 0.89 recall, and an F1 score of 0.903. One-way ANOVA confirms that the differences between tools are statistically significant (F=16.98, p¡0.001).

6.6 Contributions to Knowledge

This study makes four primary contributions to knowledge in the field of steganalysis, each aligned with one of the four research objectives.

6.6.1 First Experimentally Validated Model for Ugandan Context

This study provides the first experimentally validated steganalysis model specifically designed for the Ugandan government context. The model was developed and tested using three datasets, including 15 custom images from NITA-U operations, ensuring relevance to the local operational environment. Prior to this study, no steganalysis research had been conducted using NITA-U data or validated for Ugandan government infrastructure. This contribution directly addresses the gap identified in Objective 1 regarding applicability to the NITA-U context.

6.6.2 Complete LSB Tool with Dual Payload Handling

This study provides a complete, openly documented LSB steganography tool capable of handling both text and image payloads with configurable embedding parameters. The tool includes both encoding and decoding functionality, making it suitable for practical deployment. The source code is provided in the appendices, enabling replication and extension by other researchers. This contribution directly addresses Objective 2's requirement for a tool capable of embedding and extracting text and image payloads.

6.6.3 Comprehensive Objective Evaluation Framework

This study establishes a comprehensive evaluation framework using objective technical metrics rather than subjective user-dependent measures. The framework includes PSNR for imperceptibility measurement, SSIM for structural similarity assessment, MSE for pixel-level distortion quantification, chi-square for statistical undetectability testing, and feature detection methods (Shi-Tomasi, ORB, Harris) for resistance testing. Standard classification metrics (accuracy, precision, recall, F1) are included for detection performance evaluation. This framework provides a template for future steganalysis research and directly addresses Objective 3's evaluation requirements.

6.6.4 Comparative Benchmarking Data

This study provides comparative benchmarking data showing that custom LSB implementations can outperform commercial steganography tools in imperceptibility and undetectability. The developed tool achieved PSNR of 52.34 dB compared to 48.21 dB for Steghide, 46.73 dB for OpenPuff, and 44.15 dB for F5, with all differences statistically significant (ANOVA: F=16.98, p¡0.001). This challenges the assumption that commercial tools represent the state of the art and directly addresses Objective 4's requirement for comparative benchmarking.

6.7 Recommendations for Practice

6.7.1 Integrate Steganalysis into NITA-U Forensic Workflow

NITA-U should integrate steganalysis capabilities into its digital forensic workflow. The proposed model, which achieved 93.3% detection accuracy (Objective 4), could be deployed as a screening tool to identify images that may contain hidden data before they undergo more detailed forensic analysis. This would improve the efficiency and effectiveness of forensic investigations. The model's 93.3% detection accuracy provides a strong foundation for such deployment. This recommendation directly addresses Objective 1's requirement for applicability to NITA-U and Objective 4's validation of the proposed model.

6.7.2 Train Forensic Analysts on Detection Limitations

Forensic analysts should be trained to recognize that standard feature detection algorithms (Shi-Tomasi, ORB, Harris) cannot reliably identify LSB steganography. The em-

pirical evidence from Objective 3 (p=0.68 for Shi-Tomasi, p=0.72 for Harris, 94.7% match rate for ORB) demonstrates that reliance on these algorithms for steganalysis is likely to result in high false negative rates. Instead, analysts should use statistical methods such as chi-square analysis or the proposed model's integrated detection approach. Training should include hands-on experience with steganography tools to understand how hidden data can be embedded and detected.

6.7.3 Conduct Regular Steganalysis Audits

NITA-U should conduct regular steganalysis audits of image traffic entering and leaving government networks. These audits would help identify potential covert communication channels and data exfiltration attempts. The audits should prioritize images from high-risk sources and images that exhibit unusual file size anomalies. Given the low embedding rates used in this study (0.007-0.022%), even very small payloads should be considered suspicious. This recommendation directly addresses Objective 1's assessment of applicability to the NITA-U context.

6.7.4 Develop Formal Policies for Steganographic Content

NITA-U should develop formal policies and procedures for handling suspected steganographic content. These policies should specify when images should be subjected to steganalysis, how results should be documented, what actions should be taken when hidden data is discovered, and how evidence should be preserved for legal proceedings. The policies should also address chain of custody considerations for stego images. The validated performance of the proposed model (Objective 4) provides empirical justification for such policies.

6.7.5 Use of Custom Tools Rather Than Commercial tools.

Organizations should not rely solely on commercial steganography tools for security testing, as this study has demonstrated that custom LSB implementations can produce more undetectable stego images than commercial tools (Objective 2 findings). Security assessments should include testing with custom steganography tools to ensure that detection capabilities are robust against sophisticated adversaries. The developed tool, with its documented implementation, can serve as a reference for such testing.

6.8 Recommendations for Future Research

6.8.1 Extend to Frequency Domain Techniques

Future research should extend the evaluation to frequency domain steganography techniques including DCT and DWT-based methods. These techniques may produce different imperceptibility and detectability characteristics than spatial domain LSB. Comparative studies would help organizations select appropriate detection methods for different threat scenarios. The evaluation framework established in this study (Objective 3) could be directly applied to these techniques. This recommendation extends the scope of Objective 1's analysis of existing techniques.

6.8.2 Larger and More Diverse Datasets

Future research should evaluate the model on larger and more diverse datasets. A dataset of 1,000 or more images including JPEG-compressed images, images from social media platforms, medical images, surveillance footage, and images with different compression levels and resolutions would provide stronger evidence of generalizability. The NITA-U custom dataset should also be expanded to include more images from actual forensic cases. This would strengthen the applicability assessment required by Objective 1.

6.8.3 Adaptive Steganography Evaluation

Future research should investigate adaptive steganography techniques that selectively embed data in textured regions to avoid detection. Techniques such as HUGO (Highly Undetectable Stego) and WOW (Wavelet Obtained Weights) use content-adaptive embedding to minimize detectability (Fridrich & Kodovsky, 2012). Evaluating the proposed model against adaptive techniques would test its robustness against sophisticated adversaries and extend the feature detection resistance evaluation required by Objective 3.

6.8.4 Machine Learning Integration

Future research should explore the integration of machine learning classifiers to improve detection accuracy beyond the 93.3% achieved in this study. Ensemble methods, deep learning approaches such as CNNs, and hybrid statistical-ML models could potentially achieve higher accuracy. The feature set used in this study (PSNR, SSIM, chi-square, corner counts) could serve as input features for machine learning classifiers. This would extend the comparative benchmarking required by Objective 4.

6.8.5 Real-Time Performance Optimization

Future research should evaluate the computational efficiency of the model for real-time applications. Key metrics would include processing time per image, memory usage, throughput in images per second, and scalability to large image collections (10,000+ images). Optimizations for production deployment, including parallel processing and GPU acceleration, should be developed if necessary. This would enhance the practical utility of the tool developed in Objective 2.

6.8.6 Extension to Other Media Types

Future research should extend the model to other media types including audio and video files. Audio steganography and video steganography present different challenges and opportunities than image steganography. An integrated multimedia steganalysis system would provide comprehensive coverage for forensic applications. The detection methods developed in this study (statistical analysis, feature detection) could be adapted for other media types, extending all four objectives to new domains.

6.8.7 Operational Testing

Future research should conduct operational testing in live NITA-U environments. Real-world factors such as network latency, concurrent processing, integration with existing forensic workflows, usability by forensic analysts, and the presence of other types of data should be evaluated to prepare the model for production deployment. A pilot deployment with a small group of forensic analysts would provide valuable feedback. This would directly address Objective 1's requirement for applicability to the NITA-U context.

6.9 Limitations Acknowledged

Several limitations of this study should be acknowledged to contextualize the findings and guide future research, organized by the objective they most directly affect.

6.9.1 analysis of existing techniques

The study focused exclusively on three existing tools (Steghide, OpenPuff, F5). Other tools such as OutGuess, Hide4PGP, and Xiao Steganography were not evaluated. Additionally, only one version of each tool was tested; newer versions may have different

performance characteristics. The baseline analysis was conducted on PNG images only; JPEG, BMP, and TIFF formats were not included.

6.9.2 Development of LSB tool

The study focused exclusively on spatial domain LSB steganography. Frequency domain techniques such as DCT and DWT-based steganography were not implemented or evaluated. The developed tool uses sequential embedding without encryption; while this choice contributed to statistical undetectability, it also means the tool does not provide cryptographic security. Adaptive steganography techniques that selectively embed data in textured regions were not implemented.

6.9.3 Evaluation metrics

The dataset of 45 images, while sufficient for preliminary validation with statistical power, may not fully represent the diversity of images encountered in real-world NITA-U operations. The dataset included only PNG format images; JPEG images, which are more common on web and social media platforms, were not evaluated. Only three feature detection methods (Shi-Tomasi, ORB, Harris) were evaluated; other methods such as SIFT, SURF, and FAST were not tested.

6.9.4 Comparative benchmarking

The experiments were conducted in a laboratory environment rather than a live operational setting. Real-world factors such as network latency, concurrent processing, integration with existing forensic workflows, and analyst training requirements were not evaluated. The computational efficiency of the model for real-time applications was not systematically evaluated. The proposed steganalysis model has not been tested on live network traffic or in an operational forensic laboratory.

6.10 Conclusion

This study has demonstrated that an LSB-based steganography tool can achieve excellent imperceptibility (PSNR $>$ 52 dB; SSIM $>$ 0.997) while remaining undetectable by standard feature detection algorithms (p $>$ 0.05 for Shi-Tomasi and Harris; 94.7% match rate for ORB) and statistical chi-square analysis (chi-square ¡ 3.84). The proposed steganalysis model achieves 93.3% detection accuracy with an F1 score of 0.903, providing

a foundation for enhanced information security at NITA-U and similar organizations. Comparative benchmarking confirmed that the developed tool significantly outperforms Steghide, OpenPuff, and F5 (ANOVA: $F=16.98$; $p<0.001$).

The findings have important implications for NITA-U, the digital forensics community, and cybersecurity policy makers. NITA-U should integrate steganalysis capabilities into its forensic workflow, train analysts on appropriate detection methods, and develop formal policies for handling steganographic content. The digital forensics community should be aware that feature detection algorithms cannot reliably identify LSB steganography and should use statistical methods instead. Policy makers should include steganalysis in organizational security frameworks.

As steganographic techniques continue to evolve, the development of robust steganalysis models remains essential for protecting digital communication channels against covert exploitation. The experimental methodology, evaluation framework, and comparative results presented in this study provide a reproducible foundation for future research in this critical area of cybersecurity. The researcher hopes that this work will contribute to enhanced digital forensic capabilities in Uganda and beyond, helping security professionals detect and respond to steganographic threats that might otherwise go unnoticed.

The study concludes that LSB-based steganography, when implemented with low payload capacities and appropriate parameter selection, remains highly effective for concealing information while resisting detection. The developed steganalysis model successfully detects such hidden information with high accuracy, demonstrating that detection is possible even for well-implemented steganography. This balance between concealment and detection will continue to drive research in both fields, as each advance in steganography motivates corresponding advances in steganalysis.

The researcher extends gratitude to all who contributed to this work and hopes that the findings will serve the broader goal of enhancing cybersecurity in Uganda and beyond.

6.11 Final Summary of Objective Achievement

Table 6.1: Summary of Objective Achievement Status

Objective	**Key Evidence of Achievement**	**Status**
Objective 1	Literature review (Chapter 2); baseline analysis (Section 4.2); identification of four research gaps	ACHIEVED
Objective 2	Python LSB implementation (Section 3.4); text embedding PSNR >52 dB; image embedding recovery >99.9% (Sections 4.3, 4.4)	ACHIEVED
Objective 3	PSNR >52 dB; SSIM >0.997; chi-square <2.2; Shi-Tomasi p=0.68; Harris p=0.72; ORB 94.7% match (Sections 4.4, 4.5)	ACHIEVED
Objective 4	Comparative benchmarking (Section 4.6); 93.3% detection accuracy; ANOVA F=16.98, p<0.001	ACHIEVED

The study has successfully addressed all four objectives and answered all five research questions. The developed tools and findings provide a foundation for enhanced steganography and steganalysis capabilities at NITA-U and contribute to the broader field of digital forensics.

References

Agarwal, S., Kim, C., & Jung, K.-H. (2022). Steganalysis of context-aware image steganography using cnn. *Applied Sciences*, *12*(21), 10793.

Bas, P., Filler, T., & Pevný, T. (2011). "break our steganographic system": The ins and outs of organizing boss. *Information Hiding - 13th International Conference, IH 2011*, *6958*, 59–70. https://doi.org/10.1007/978-3-642-24178-9_5

Binghamton University. (2024). *Bossbase v1.01*.

Boroumand, M., Chen, M., & Fridrich, J. (2019). Deep residual network for steganalysis. *IEEE Transactions on Information Forensics and Security*, *14*(5), 1255–1270.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Byun, H., Kim, J., Jeong, Y., Seok, B., Gong, S., & Lee, C. (2024). Security analysis of cryptocurrency wallets against brute-force attacks. *Electronics*, *13*(13), 2433.

Chaganti, R., Ravi, V., Alazab, M., & Pham, T. D. (2021). Stegomalware: A systematic survey. *arXiv preprint arXiv:2110.02504*.

Chan, C.-K., & Cheng, L. M. (2004). Hiding data in images by simple lsb substitution. *Pattern Recognition*, *37*(3), 469–474. https://doi.org/10.1016/j.patcog.2003.08.007

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Image steganography: A review. *Signal Processing*, *90*(8), 2334–2357.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Farooq, N., & Selwal, A. (2023). Image steganalysis using deep learning: A systematic review. *Journal of Ambient Intelligence*, *14*(6), 7761–7793.

Fridrich, J. (2009). *Steganography in digital media*. Cambridge University Press.

Fridrich, J., Goljan, M., & Du, R. (2001). Attacking the outguess steganographic system. *Information Hiding Workshop*.

Fridrich, J., & Kodovsky, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, *7*(3), 868–882.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. *Alvey Vision Conference*, 147–151.

Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, *44*(13), 800–801. https://doi.org/10.1049/el:20080522

Johnson, N. F., & Jajodia, S. (1998). Steganography: Seeing the unseen. *IEEE Computer*, *31*(2), 26–34.

Ker, A. D. (2004). Improved detection of lsb steganography in grayscale images. *International Workshop on Information Hiding*, 97–115. https://doi.org/10.1007/978-3-540-30114-1_8

Ker, A. D. (2013). Moving steganography and steganalysis from the laboratory to the real world. *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '13)*, 45–58. https://doi.org/10.1145/2482513.2482530

Kharrazi, M., Sencar, H. T., & Memon, N. (2004). Image steganography: Concepts and practice. *Lecture Notes Series, Institute for Mathematical Sciences*, 35–49.

Kheddar, H., Hemis, M., Himeur, Y., Megias, D., & Amira, A. (2024). Deep learning for steganalysis: A review. *Neurocomputing*, *589*, 127528.

Luo, W., Wei, K., Li, Q., Ye, M., Tan, S., Tang, W., & Huang, J. (2024). A comprehensive survey of digital image steganography and steganalysis. *APSIPA Transactions*, *13*(1).

Mielikainen, J. (2006). Lsb matching revisited. *IEEE Signal Processing Letters*, *13*(5), 285–287. https://doi.org/10.1109/LSP.2006.870357

NITA-U. (2023). *Annual report 2022-2023*.

OpenPuff. (2024). *Openpuff steganography tool*.

Owolabi, O., & Akintola, K. (2022). Cybersecurity challenges in sub-saharan africa. *African Journal of Information Security*, *8*(2), 45–62.

Patel, R., & Sharma, S. (2021). Comprehensive survey of image steganography. *Multimedia Tools and Applications*, *80*, 25067–25108.

Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding: A survey. *Proceedings of the IEEE*, *87*(7), 1062–1078.

Pevný, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, *5*(2), 215–224. https://doi.org/10.1109/TIFS.2010.2045842

Provos, N. (2001). Defending against statistical steganalysis. *10th USENIX Security Symposium*, 323–335. https://www.usenix.org/legacy/events/sec01/provos.html

Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, *1*(3), 32–44. https://doi.org/10.1109/MSECP.2003.1203220

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. *ICCV*, 2564–2571.

Schaefer, G., & Stich, M. (2003). Ucid: An uncompressed color image database. *Storage and Retrieval Methods and Applications for Multimedia 2004*, *5307*, 472–480. https://doi.org/10.1117/12.525375

Selvaraj, A., Ezhilarasan, A., Wellington, S. L. J., & Sam, A. R. (2021). Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning. *IET Image Processing*, *15*(2), 504–522.

Shi, J., & Tomasi, C. (1994). Good features to track. *CVPR*, 593–600.

Singh, P., Sharma, A., & Kumar, R. (2022). Image steganography and steganalysis: A review. *International Journal of Information Security*, *21*, 123–145.

Steghide. (2024). *Steghide user manual*.

Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image steganography: A review of the recent advances. *IEEE Access*, *9*, 23409–23423.

Uganda National CERT. (2022). *National cybersecurity report*.

USC. (2024). *Usc-sipi image database*.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, *2*, 1398–1402. https://doi.org/10.1109/ACSSC.2003.1292216

Westfeld, A. (2001). F5—a steganographic algorithm. *Information Hiding Workshop*, 289–302.

Westfeld, A., & Pfitzmann, A. (2000). Attacks on steganographic systems. *International Workshop on Information Hiding*, 61–76. https://doi.org/10.1007/10719724_5

Xu, G., Wu, H.-Z., & Shi, Y. Q. (2016). Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, *23*(5), 708–712. https://doi.org/10.1109/LSP.2016.2548421

Yedroudj, M., Comby, F., & Chaumont, M. (2020). Yedroudj-net: A deep learning approach for image steganalysis. *IEEE Transactions on Information Forensics and Security*, *15*, 2589–2601.

Yu, L., Zhao, Y., Ni, R., & Li, T. (2010). Improved adaptive lsb steganography based on chaos and genetic algorithm. *EURASIP Journal on Advances in Signal Processing*, *2010*, 1–6. https://doi.org/10.1155/2010/876946

Zhu, Y., Wang, X., Chen, H.-S., Salloum, R., & Kuo, C.-C. J. (2023). Green steganalyzer: A green learning approach. *APSIPA Transactions*, *12*(1).