Steganalysis for Data Loss Prevention

RP24_25J_003

Final Report

IT21229220-G.B.T.G INDRAJITH

B.Sc. (Hons) Degree in Information Technology specialized in

Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology

Feb 2025

Leveraging Machine Learning for Data Loss Prevention

Project ID - 24-25J-
003

G.B.T.G Indrajith

(IT21229220)

Supervisor – Mr. Amila Senarathne

Co-Supervisor – Ms. Suranjini silva

B.Sc. (Hons) in Information Technology Specializing in Cyber Security

Department of Information Technology Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DECLARATION

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Group member name	Student ID	Signature
Indrajith G.B.T.G	IT21229220	

The above candidate is carrying out research for the undergraduate Dissertation under supervision of the undersigned.

…………………………………..

……………………………….

Signature of the supervisor Date

(Mr. Amila Senerathna)

…………………………………..

………………………………...

Signature of Co-supervisor Date

(Ms. Suranjini Silva)

ACKNOWLEDGEMENT

I extend my sincere gratitude to my supervisor, Mr. Amila Senerathna, and co-supervisor, Ms. Suranjini Silva, for their invaluable guidance and support throughout this research study. I'm thankful to industry experts for sharing their expertise which helped me get a certain domain knowledge. Special thanks to my team members for their contributions, and to those who aided me willingly. Lastly, my heartfelt appreciation to my family for their constant love, assistance, and encouragement.

ABSTRACT

Digital communication expansion has led to an increased threat of unauthorized data transfer through image-based steganographic strategies. Existing Data Loss Prevention (DLP) tools do not recognize embedded transmissions even when they exist beneath traditional data security measures. The proposed research combines steganalysis algorithms with Optical Character Recognition (OCR) and sensitive data identification functions to form an exhaustive Data Loss Prevention (DLP) framework which detects and blocks unauthorized leakage within image files.

The system base implements steganalysis technologies that feature Improved CNN model detection of S-UNIWARD and WOW algorithms and performs Least Significant Bit analysis for simple embedding detection. Multiple voting among ensemble systems serves to enhance the detection accuracy across different stego types. The system utilizes Tesseract OCR supported by multiple preprocessing methods which extracts text content from images while maintaining compatibility with diverse file types and image qualities.

The analysis system employs sensitivity guidelines to determine Personally Identifiable Information (PII) using rule-defined thresholds. Whenever a violation occurs the system creates a risk score that blocks the image transmission through integrated platforms which include email clients. The demo web application presents live monitoring enforcement features which security officers can access through their admin dashboard.

Through its sensitivity guidelines the analysis system defines rules for threshold values to identify Personally Identifiable Information (PII). The system generates a risk score after a violation which blocks image transfers through integrated platforms that include email clients. The demo web application provides security officers with access to enforcement monitoring features through their admin dashboard.

Keywords: Data Loss Prevention, Steganalysis, OCR, LSB Detection, Image Security, Sensitive Data Classification, Spatial domain steganalysis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
AUC	Area Under the Curve
API	Application Programming Interface
CCPA	California Consumer Privacy Act
CER	Character Error Rate
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DLP	Data Loss Prevention
FNR	False Negative Rate
FPR	False Positive Rate
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IOS/IEC	International Organization for Standardization/International Electrotechnical Commission
LSB	Least Significant Bit
NIC	National Identity Card
OCR	Optical Character Recognition
OEM	OCR Engine Mode
PII	Personally Identifiable Information
PSM	Page Segmentation Mode
RST	Representational State Transfer
ROC	Receiver Operating Characteristic

INTRODUCTION

Background

Organizations throughout every business sector maintain data protection as their main focus because of today's digitally connected business structure. The rapid growth of data transmission through cloud services and email communication together with remote collaboration tools has made data exposure risks much more severe. Enterprise cybersecurity architecture depends on Data Loss Prevention (DLP) systems to prevent unauthorized data transmissions that threaten financial and legal as well as reputational harm to organizations. Modern DLP systems monitor structured data but their capabilities now also cover unstructured data that appears in documents emails and images because these formats have become key targets for cyber attackers to move their stolen information undetected. The implementation of effective DLP strategies has turned into both a legal requirement and a technical necessity because of strict penalties imposed by privacy regulations such as GDPR and CCPA. Gartner indicates that organizations with no defined Data Loss Prevention (DLP) strategy face elevated dangers from internal threats and unintentional data loss particularly in sectors including finance, healthcare and government sectors [1]. Research reveals that enterprises now identify DLP as crucial for their cloud security position because sensitive data continues to spread across hybrid and multiple cloud platforms [2].

Fundamentally sophisticated yet invisible data exfiltration occurs through image-based steganography technology which secures confidential data inside normal-looking image files. Data leakage using steganography bypasses traditional security rules since the method embeds hidden messages inside digital media files which signature-based detection systems frequently miss. The LSB modification technique combines with transform domain embedding while using adaptive steganographic algorithms S-UNIWARD and WOW to create challenges in identifying contaminated image files from ordinary ones. The images remain invisible to standard DLP alerts while users can readily distribute these files through email and social networks and storage platforms. The detection of data theft becomes impossible because attackers exploit this vulnerability to move data undisguised through security frameworks. The advancement of steganographic tools which use deep learning for resistance against forensic detection attracts APT actors as stated in Fridrich and Kodovsky's study [3]. New threat intelligence reports document that attackers use real-world cyberattacks by placing confidential data within images and transmitting them undetected from corporate networks [4].

The development of covert channels in cybersecurity fields added intricate challenges for security teams who aim to stop unauthorized data extraction. The intentional disguise of transmitted data occurs through covert channels which use legitimate systems' unexpected features to hide communications. The use of image-based steganography in conjunction with OCR bypass techniques has created a dangerous development that allows operators to invisibly transmit sensitive text through images across communication channels. Steganographic algorithms embed confidential text into images for later extraction by OCR engines which prevents traditional content inspection as well as DLP mechanisms from detecting plaintext streams. Since these obfuscated channels utilize multiple levels of concealment they become very difficult to detect because visual disguise combines with semantic evasion.

APT groups have incorporated this approach because they insert hidden text or documents inside ordinary images which can be extracted by OCR-enabled scripts after transfer completion. The techniques have been used by specific attack campaigns as reported by Symantec for extracting source code and credentials from breached environments without detection by endpoint security tools [5]. Adversaries implement image preprocessing steps including distortion and compression according to Liu et al.'s research to reduce OCR detection but maintain readable text outcomes at recipient locations [6]. The combination of Tesseract OCR and open-source steganographic libraries makes it more difficult to detect covert channels because these tools enable strong attack methods which are undetectable [7]. Modern security infrastructure requires multi-modal DLP systems able to perform steganalysis and semantic text analysis simultaneously because of recent attacks.

Problem Domain

Enterprise security depends heavily on the ability to manage outbound data because modern organizations use digital communication for everyday operations. DLP systems designed for traditional data protection fail to properly scan hidden information contained inside image files and structured text and document metadata. Detection of anomalous content in image-based data exfiltration proves challenging for standard scanning tools since the complex structure and visuals of format files generate high rates of false positives. Criminals use three main techniques based on Least Significant Bit (LSB) embedding, adaptive steganography and deep learning-based steganographic methods to embed sensitive data in images while maintaining their visual integrity. The embedded content can transfer through email and cloud-based platforms after bypassing conventional security measures established at perimeter edges. Reliable steganography techniques now include the use of encryption for stego-payloads and hidden text that becomes readable only through Optical Character Recognition engines to make detection even more challenging. Many DLP systems have a critical weakness when dealing with data hiding techniques because they lack the ability to analyze images in real time which emphasizes the need for advanced systems that handle image steganalysis and semantic content processing. The solution to this problem demands multiple detection methods which unite image forensic techniques with machine learning models together with text recognition methods for discovering and blocking attempted covert data transfers.

Traditional Data Loss Prevention (DLP) tools detect and stop the leakage of structured data including credit card numbers, email addresses and database exports with success. Traditional Data Loss Prevention tools show limited effectiveness when analyzing the modern, sophisticated methods of exfiltrating data through unstructured formats that include images and videos and multimedia content. The majority of current DLP systems do not have proper defense against covert data channels such as steganography and OCR-embedded text in images because they work through pattern matching and keyword scanning and rule-based inspection. Such security tools lack the capability to detect hidden information inside image pixels or non-selectable graphical text objects which produces vulnerabilities within security systems. The analytical abilities of traditional DLP systems fall short when they need to evaluate data hidden in complicated file arrangements and visual content structures because they lack enriched content examination based on image forensic analysis and semantic understanding. During routine scans enterprises become vulnerable to insider threats and advanced persistent threats because attackers use evasive low-signal approaches that traditional detection techniques fail to detect. Data leak incidents affecting more than 60% of sensitive information occur with unclassified file types which traditional DLP solutions cannot effectively identify or block [8]. DLP systems produce false alarm rates that remain high because their heuristics are outdated leading organizations to overlook valid risks because user satiation with false alerts causes threat detection failure [9].

Research Gap

Modern Data Loss Prevention frameworks suffer from a fundamental flaw because they lack proper integration between steganalysis and Optical Character Recognition (OCR) capabilities. Modern DLP tools provide either textual content inspection or image metadata analysis but their functionality does not support simultaneous examination of these elements. These systems lack the capability to find multi-layered threats when sensitive textual information resides in steganographic images as they get extracted through OCR processes. Current enterprise software solutions do not merge steganalysis tools, which operate in both academic and forensics domains and OCR engines used for document digitization to provide real-time content-based image inspection capabilities. The division of security functions creates an opportunity for adversaries to hide restricted information including personal identifiers and classified text inside images before sending them across authorized channels. Security gaps in current infrastructure result from the inability of steganographic detection models to communicate with OCR-driven classification tools. Experts warn about developing unified DLP solutions that must identify hidden information and visually embedded text because hybrid exfiltration methods have become more advanced [10].

Research Problem

The modern digital environment features sophisticated threats of data extraction which occurs through elaborate covert and non-standard methods. The hiding of sensitive material inside image files functions as one standard technique for adversaries who want to conceal their information. Steganography combined with image pixel encryption allows adversaries to hide confidential data beyond conventional Data Loss Prevention tools by utilizing image formatting methods for keyword evasion. Enterprise-level DLP systems encounter an intensive challenge because they need to develop methods to ensure effective prevention and detection of hidden sensitive information in image files. The detection challenge increases because images come in different formats and hiding techniques are subtle and text detection through OCR produces diverse results concerning both quality and placement. The standalone operation of steganalysis tools and OCR engines prevents their use for enterprise-level applications because they do not support real-time processing or automation. Designing and deploying a single DLP tool emerges as the central research initiative because it brings machine learning steganalysis and intelligent OCR text scanning together to check image data flow out from the system and stop sensitive data transmission. Solving this issue demands technical development of multiple detection techniques within a scalable framework that operates effectively in high-volume systems like enterprise email and secure file-sharing systems.

Research Objectives

Main Objective

The main objective of this research project is to **design and implement a Data Loss Prevention (DLP) system that integrates steganalysis, Optical Character Recognition (OCR), and sensitive data classification mechanisms to detect and block unauthorized data transmission through image files**. This unified approach aims to address the growing challenge of image-based data exfiltration, where adversaries embed confidential information within digital images using steganographic techniques or render sensitive text that is only retrievable through OCR processes. The proposed system will employ an improved Convolutional Neural Network (CNN)-based steganalysis model to identify covert data hidden within image structures, while leveraging OCR technologies to extract and analyze any visible or embedded text content. The extracted data will then be passed through a sensitivity classification engine to evaluate whether it contains Personally Identifiable Information (PII) or other confidential material. By combining these three components—**image forensics, text extraction, and contextual classification**—the system seeks to provide a comprehensive solution capable of inspecting outbound image data in real time and enforcing policy-based blocking when violations are detected. This objective not only contributes to advancing the technical capabilities of enterprise DLP systems but also supports compliance with data protection regulations by proactively mitigating the risk of covert data leaks.

Sub-Objectives

To achieve the overarching goal of developing a comprehensive DLP system capable of detecting and blocking unauthorized data exfiltration through image files, the project has been broken down into the following four sub-objectives:

1. Detect steganographic content using Improved CNN models and LSB detection models.

   This sub-objective focuses on designing a deep learning-based steganalysis engine capable of identifying concealed data embedded using both simple (e.g., LSB) and advanced (e.g., WOW, S-UNIWARD) steganographic techniques. The model aims to distinguish between clean and stego images with high accuracy while maintaining performance efficiency suitable for real-time applications.

2. Integrate Optical Character Recognition (OCR) techniques for extracting text from image files.

   This objective involves the use of the Tesseract OCR engine combined with various image preprocessing methods (e.g., binarization, noise reduction) to improve the accuracy of text extraction across different image formats and resolutions. The goal is to detect sensitive content visually embedded in images, whether typed, printed, or stylized.

3. Design a sensitivity-based classification engine for analyzing and categorizing extracted textual content.

   This component will classify OCR-extracted text based on configurable sensitivity thresholds to identify potential Personally Identifiable Information (PII), corporate confidential data, or compliance-related keywords. The engine will trigger alerts or block actions when thresholds are met or exceeded.

4. Implement a fully integrated DLP pipeline with a demo email client and security dashboard.

   The final objective is to build a working web-based prototype that demonstrates the real-time integration of steganalysis, OCR, and classification. It will include a secure email client interface capable of inspecting outgoing image attachments and an administrative dashboard for monitoring violations and system activity.

METHODOLOGY

System Architecture Overview

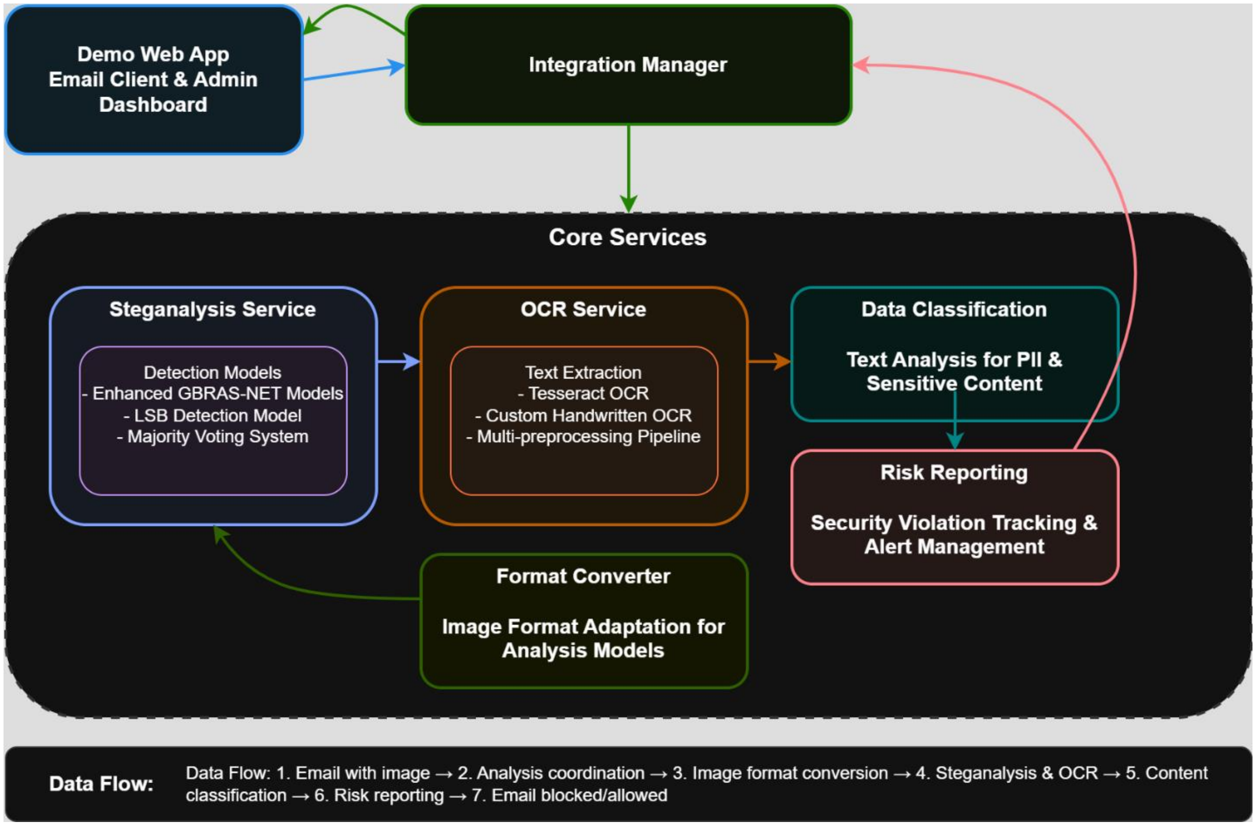

Demo Web App
Email Client & Admin Dashboard
Integration Manager
Core Services
Steganalysis Service
Detection Models
- Enhanced GBRAS-NET Models
- LSB Detection Model
- Majority Voting System
OCR Service
Text Extraction
- Tesseract OCR
- Custom Handwritten OCR
- Multi-preprocessing Pipeline
Data Classification
Text Analysis for PII & Sensitive Content
Risk Reporting
Security Violation Tracking & Alert Management
Format Converter
Image Format Adaptation for Analysis Models
Data Flow:
Data Flow: 1. Email with image → 2. Analysis coordination → 3. Image format conversion → 4. Steganalysis & OCR → 5. Content classification → 6. Risk reporting → 7. Email blocked/allowed

Figure 1:- DLP steganalysis System Architecture

The diagram illustrates the high-level architecture of your DLP Steganalysis system, showing how different components interact to detect and prevent data exfiltration.

Key Components

1. Demo Web App
    - Provides user interfaces for email communication and security monitoring.
    - Contains both the email client for users and the admin dashboard for security personnel.
    - Serves as the entry point for email communications that will be analyzed.
2. Integration manager
    - Orchestrates communication between all services.

- Manages service health and startup/shutdown.
- Routes image data through appropriate analysis pipelines.
- Makes final decisions on whether to block or allow communications

3. Core services
    a. Steganalysis Service.
    - Analyzes images for hidden data (steganography).
    - Uses enhanced GBRAS-NET models for detecting sophisticated steganography techniques.
    - Implements LSB (Least Significant Bit) detection for simpler steganography.
    - Employs a majority voting system to improve detection accuracy and reduce false positives.

    b. OCR Service
    - Extracts text from images using a hybrid approach.
    - Integrates Tesseract OCR for baseline text recognition.
    - Incorporates your custom OCR model trained on handwritten data for improved recognition.
    - Applies multiple preprocessing techniques to optimize text extraction.

    c. Data Classification Interface
    - Analyzes extracted text to identify personally identifiable information (PII).
    - Classifies content based on sensitivity levels.
    - Determines if text content violates data protection policies.

    d. Risk Reporting Module
    - Records security violations for audit purposes.
    - Manages alerting and notification systems.
    - Provides risk scoring for different types of violations.

    e. Format Converter
    - Transforms images into formats compatible with different analysis tools.
    - Ensures optimal preprocessing for both steganalysis and OCR.

Data Flow

1. User attempts to send an email with an image attachment through the web interface.
2. Integration Manager receives the request and coordinates the analysis.
3. Format Converter prepares the image for analysis by both steganalysis and OCR services.
4. Steganalysis Service examines the image for hidden data using multiple detection models.
5. OCR Service extracts any visible text from the image.
6. Data Classification analyzes the extracted text for sensitive information.
7. Risk Reporting records any detected violations.
8. Integration Manager makes a final decision to block or allow the email based on analysis results.
9. User receives feedback on whether their email was sent or blocked.

Steganalysis Module

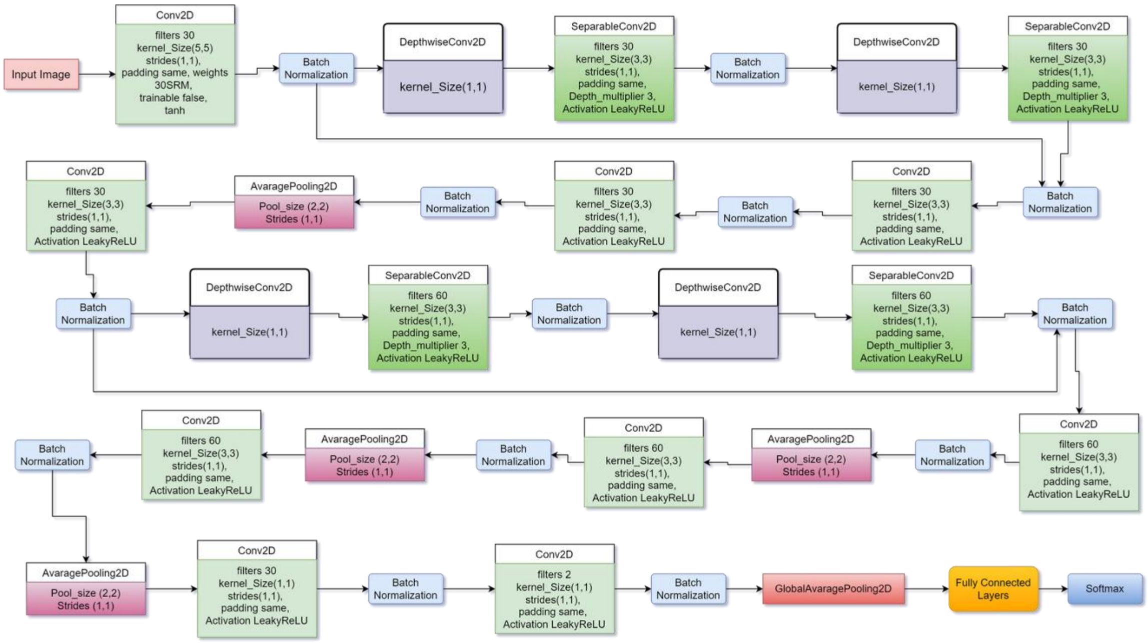

Input Image
Conv2D
filters 30 kernel_Size(5,5) strides(1,1), padding same, weights 30SRM, trainable false, tanh
Batch Normalization
DepthwiseConv2D
kernel_Size(1,1)
SeparableConv2D
filters 30 kernel_Size(3,3) strides(1,1), padding same, Depth_multiplier 3, Activation LeakyReLU
Batch Normalization
DepthwiseConv2D
kernel_Size(1,1)
SeparableConv2D
filters 30 kernel_Size(3,3) strides(1,1), padding same, Depth_multiplier 3, Activation LeakyReLU
Batch Normalization
Conv2D
filters 30 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
Conv2D
filters 30 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
AvaragePooling2D
Pool_size (2,2) Strides (1,1)
Conv2D
filters 30 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
DepthwiseConv2D
kernel_Size(1,1)
SeparableConv2D
filters 60 kernel_Size(3,3) strides(1,1), padding same, Depth_multiplier 3, Activation LeakyReLU
Batch Normalization
DepthwiseConv2D
kernel_Size(1,1)
SeparableConv2D
filters 60 kernel_Size(3,3) strides(1,1), padding same, Depth_multiplier 3, Activation LeakyReLU
Batch Normalization
Conv2D
filters 60 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
AvaragePooling2D
Pool_size (2,2) Strides (1,1)
Conv2D
filters 60 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
AvaragePooling2D
Pool_size (2,2) Strides (1,1)
Conv2D
filters 60 kernel_Size(3,3) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
AvaragePooling2D
Pool_size (2,2) Strides (1,1)
Conv2D
filters 30 kernel_Size(1,1) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
Conv2D
filters 2 kernel_Size(1,1) strides(1,1), padding same, Activation LeakyReLU
Batch Normalization
GlobalAvaragePooling2D
Fully Connected Layers
Softmax

Figure 2: Improved Steganlaysis CNN model Architecture

The DLP system presents a steganalysis module that detects hidden data inside image files by utilizing S-UNIWARD and WOW advanced methods for steganographic algorithms. The algorithms remain widespread in steganographic studies because they accomplish high data insertion while maintaining minimal statistical alterations and thus prove difficult to detect with traditional methods. Steganalysis requires an improved Convolutional Neural Network (CNN) model built for spatial-domain work which the system incorporates for detection purposes.

The CNN model conducts training using datasets which contain image pairs of cover images and stego images created through S-UNIWARD and WOW techniques. These layers enhance the detection of geometrical patterns and imperceptible embedding artifacts which neither humans nor pixel-based methods can detect. This model uses batch normalization in combination with residual connections and dropout regularization for faster training along with reduced overfitting. The framework contains updated features to distinguish between neat images along with LSB-embedded images and S-UNIWARD and WOW-altered images within a multi-class classification system.

The trained CNN functions through RESTful API by accepting image files to process through the detection pipeline that returns probability scores about steganographic content likelihood. The system utilizes a majority voting process so multiple detection models operate together to enhance accuracy while cutting down false positive results. The enhanced CNN-based steganalysis module when integrated into the DLP system provides strong capabilities to detect advanced secret communication methods through images thus becoming a vital building block for full data leak prevention practices.

The Least Significant Bit (LSB) detection pipeline operates as a key component of steganalysis modules by seeking basic steganographic methods which modify digital images' pixel value least significant bits. Through LSB steganography malicious actors embed secret data into images by modifying the least significant bit of pixel values which results in minimal perceptible changes to picture appearance. Steganography adopts this simple method because of its large capacity storage and straightforward implementation process.

The LSB detection pipeline starts its analysis at pixel level by reviewing statistical inconsistencies between patterns found in grayscale or RGB image channels. The system undertakes preprocessing to separate LSB planes by stripping the least significant bit from all pixel values throughout each color channel. Anomaly detection in LSB layers utilizes statistical assessments of both bit-plane complexity and local smoothness as well as noise distribution patterns. Random patterns appear prominently in the LSB planes of clean images whereas stego images reveal structured or biased patterns throughout their LSB planes because they hold embedded data.

The detection mechanism of the pipeline uses Chi-square analysis together with RS (Regular-Singular) steganalysis techniques to measure deviations in pixel pair distributions versus their natural image counterparts. The detection of abnormal LSB value regularities is done through two methods that include entropy measures coupled with histogram comparisons. The manually designed features enter into lightweight support vector machine (SVM) or decision tree classifiers for distinguishing between clean and stego images through signatures of LSB changing.

The pipeline generates a confidence score together with binary classification labels that are joined with the results of complex detectors such as WOW or S-UNIWARD detection systems through majority voting rules. The combined usage of analysis models increases steganalysis service reliability thus enabling detection of complex and basic steganographic attacks.

The proposed DLP system uses a majority voting mechanism to combine detection models results in order to increase the reliability of steganographic content detection. There exists no detection algorithm capable of achieving consistent high accuracy because various steganographic techniques manipulate image structures through different methods that include LSB, S-UNIWARD and WOW. Using a single classifier creates the potential for wrong negative outcomes and diminished prediction capability across different classes.

Independent classifiers using the majority voting mechanism operate at the model level to create an ensemble approach where each component makes separate binary clean or stego predictions about the selected input image. The final prediction results from combining individual decisions into one statement and using the most frequently chosen class label. A decision of "stego" is reached when two out of three models identify hidden data while the third model predicts clean.

This approach has several advantages. Such an approach decreases model-specific prediction errors since weak model predictions can be corrected by consensus from more accurate ones. The method achieves broad format and steganography type generalization through utilization of different detectors that specialize in particular domains. The voting scheme enables weighted voting that enhances model influence by assigning higher vote weights to detectors according to their recorded past validation results.

The DLP system uses the majority voting logic across its steganalysis service layer to scrutinize all incoming and outgoing images through a multi-layer analysis process. The system performs block or OCR qualification procedures using results calculated from allStageEx results. The combination of multiple decisions through this strategy increases detection precision while it reduces incorrect alerts and enhances the system's capability to respond to various security challenges.

OCR component

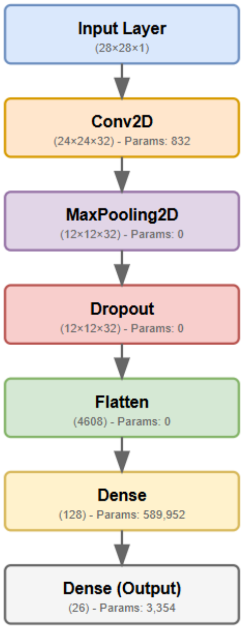


Figure 3: OCR model architecture

The developed Detection and Leakage Prevention (DLP) system includes Tesseract OCR as an open-source software for Optical Character Recognition that Google produces. Tesseract stands out among other tools because it supports numerous languages and demonstrates adaptability to extract text from many diverse image formats. The system integration enables automatic text recognition and extraction from outbound image files thus bypassing traditional text filters in DLP systems.

The OCR process starts after an image completes the steganalysis step of the pipeline. Images go to the OCR service for processing either because analysis results show non-steganographic characteristics or because the system configures universal image screening. The preprocessing operations begin before text extraction because they enhance OCR accuracy levels. The preprocessor performs grayscale conversion and then applies thresholding using Otsu's method and follows with noise reduction and image normalization steps. The application uses OpenCV to process images with poor resolution by applying extra filters including median blurring and contrast enhancement procedures.

Tesseract is invoked with custom configuration flags to optimize its performance, such as specifying language packs, defining page segmentation modes (--psm), and using **OCR Engine Mode (OEM)** settings that determine whether to use legacy, LSTM, or combined engines. The output is a raw text stream, which is then passed through **post-processing routines** to remove artifacts (e.g., newline characters, OCR misreads) and standardize the text format.

The system accepts PNG JPEG BMP and TIFF formats while executing multiple attachment processing through its batch transaction capabilities. The data classification module receives extracted text so it can analyze it through sensitivity thresholds along with predefined keyword patterns like emails as well as ID numbers and company secrets. When the system detects a policy violation it has the ability to flag or stop transmissions directly in real-time.

Tesseract OCR functionality integrated within the DLP workflow enables the system to detect image-based data because it evaluates both screenshots and visual text embedded in images. This enhancement strengthens the defense system's capacity to stop data leakage attempts by using image-based hiding methods which operate as essential elements for adequately protecting organizations.

OCR accuracy depends heavily on effective preprocessing techniques when dealing with real images that show signs of noise and distortions or poor lighting. Setups in the proposed DLP system perform image data preparation through preprocessing methods preceding Tesseract OCR engine receipt. Three optimization methods combine to enhance text readability and minimize errors in Optical Character Recognition through the processes of noise reduction along with binarization and contrast enhancement.

Background structures and scanning errors as well as compression artifacts cause image noise that prevents OCR from recognizing the text correctly. The OpenCV library enables the system to reduce noise through its application of denoising filters that include median blur and Gaussian blur. The median filtering technique efficiently maintains edges during its process of eliminating salt-and-pepper noise which frequently occurs in scanned documents and screenshots. Before grayscale conversion the system applies denoising to individual color channels to maintain uniform noise reduction for all images.

An OCR engine requires text-background separation so the image conversion through binarization helps transform images into black and white binary format. The system employs adaptive thresholding methods with Otsu's method to find the perfect threshold value for transforming images to binary format. The computation of threshold values by adaptive methods becomes advantageous specifically in images showing lighting inconsistencies or background irregularities because these methods take local rather than global measurements. The resulting text regions become much easier to see because this step lowers the chances of false character detection.

OCR systems experience significant challenges when trying to correctly process images which exhibit low contrast because of factors like document degeneration or screenshots taken under dim lighting conditions. The system utilizes contrast enhancement methods which implement both histogram equalization and Contrast Limited Adaptive Histogram Equalization (CLAHE). The techniques reallocate pixel brightness which provides better text-and-background separation so text characters appear more distinguishable for straightforward extraction.

The pre-processing methods execute one after the other before OCR based on image quality evaluations. Their sequential execution produces a strong preprocessing system which delivers high accuracy no matter what format or quality of input is provided. The strategies integration within the OCR module enhances its capacity to extract useful text data from pictures across demanding practical usage contexts where screenshots, scans or compressed file images exist.

The proposed Data Loss Prevention (DLP) system uses its OCR component to extract text from various image formats used in real-world enterprise situations including JPEG (JPG), Portable Network Graphics (PNG), Tagged Image File Format (TIFF), Bitmap (BMP) as well as Scalable Vector Graphics (SVG). During practical use sensitive content embedded in images is found within different file types because users behave differently and because image compression tools and email attachment formats exist. For a robust and adaptable DLP system it is vital to achieve text extraction from different image file types.

The OCR processing starts by detecting MIME types and corresponding file extensions of incoming images. A format handler layer through libraries Pill (PIL) and OpenCV performs the conversion process of standardized images into uniform internal grayscale or RGB formats regardless of original encoding. Standardization through conversion enables one to maintain stable processing together with OCR application across multiple image formats.

Each format presents unique technical challenges:

- **JPEG** images are often lossy and may introduce compression artifacts that distort character edges.
- **PNG** supports lossless compression and alpha transparency, which may require alpha channel removal before processing.

- **TIFF** files, commonly used in document scanning, may contain multiple pages or layers, necessitating page iteration and selective frame extraction.
- **BMP** images are uncompressed and large in size, often requiring resizing to meet performance thresholds.
- **SVG** files, while vector-based, may embed raster images or text layers and are preprocessed through rasterization before OCR.

The system tackles these problems through predefined correction protocols for different document formats and automatic back-up procedures. The system processes TIFF files through functions provided by the Python Imaging Library then extracts multiple frames while the PNG transparency features receive neutral background flattening operations. The preprocessing steps aim to create image normalization which produces reliable and high-quality input data for the OCR engine known as Tesseract.

The DLP system achieves comprehensive analysis by handling different image formats that enables smooth processing of varying content users produce so it operates effectively regardless of the wide range of format variations across different devices and platforms. The system capabilities prove vital for standardizing image input primarily in enterprise environments where users frequently share unstandardized documents and screenshots.

Data Classification

The Sensitivity Rule Engine of the proposed DLP system serves as a primary mechanism to detect and categorize textual data extracted from picture files regarding both Personally Identifiable Information (PII) along with secret attributes. A builtin functionality allows the system to perform regex pattern and sensitivity rule evaluation of OCR results allowing it to detect and prevent sensitive content from passing through image transmissions.

The rule engine operates as a modular pipeline, where each rule is defined to detect specific data categories such as:

- **Email addresses**: Pattern matching strings with format username@domain.com using regex such as [\w\.-]+@[\w\.-]+\.\w+.
- **Phone numbers**: Recognizing country-specific formats, e.g., (\+94)?\s?\d{9,10} for Sri Lankan mobile numbers.
- **National Identity Numbers**: Matching patterns like Sri Lankan NIC numbers (\d{9}[VvXx]|\d{12}).
- **Financial identifiers**: Credit card numbers, bank account digits, IBAN codes, etc.
- **Location or address data**: Partial or full address components based on known patterns or lookup dictionaries.

A match receives its sensitivity score through evaluation of both its content and context data points and document type. The risk score of an individual email address remains lower while a single image containing name along with NIC and contact number will trigger an automated violation check when the risk threshold exceeds predefined parameters. Organizations can automate their policy response through this risk scoring approach since they can set different response actions according to the score level ranging from monitoring to stopping the image.

The engine enhances accuracy and decreases false positives through context-aware detection which confirms regex hits by proximity rules and dictionary-based cross-verification processes. The system allows extension of its rule set through flexible administration controls that enable organizations to add their specific keywords and document identifiers for scanning purposes.

The integration of a rule engine within the classification stage enables the DLP system to convert unprocessed OCR output into identifiable valuable information for better detection of unauthorized data exposure from images. The system's compliance with GDPR alongside other data protection rules becomes possible through this approach which also enables security teams to detect sensitive information before it is leaked.

The proposed DLP system achieves flexible and policy-based enforcement of sensitive data leakage through its Data Classification module which includes threshold configuration options and violation triggers. The system provides organizations with a mechanism to customize sensitivity levels which adapts to their operational requirements and risk management criteria along with compliance necessities.

OCR-extracted text goes through the Sensitivity Rule Engine to check if it matches defined patterns including Personally Identifiable Information (PII) along with confidential keywords. The platform generates sensitive risk scores for every identified matched entity (email address, phone number, national ID, financial data) while producing cumulative measurements for the complete document.

The **configurable threshold** represents the minimum sensitivity score at which a particular action should be triggered. For example, an organization may define thresholds as follows:

- **Low Risk (Score 1–3)**: Log the incident for future auditing.
- **Medium Risk (Score 4–6)**: Notify the user or system administrator.
- **High Risk (Score 7 and above)**: Immediately block the transmission and generate a compliance alert.

The threshold-based classification system allows administrators to calibrate behavior at various risk levels while reducing false alarms while protecting important disclosures. Security teams can modify thresholds by using an administrative interface without requiring any changes to the core system code.

Violation triggers are associated with these thresholds and define the system's response when a sensitivity level is met or exceeded. Triggers can initiate various actions, such as:

- **Blocking** the outbound image from being sent via email or uploaded.
- **Alerting** the security team through a dashboard notification or email.
- **Logging** the incident with full metadata (timestamp, user ID, extracted content, image hash) for future review or forensic analysis.
- **Escalating** based on the role of the user or the type of content detected.

The threshold-based classification system allows administrators to calibrate behavior at various risk levels while reducing false alarms while protecting important disclosures. Security teams can modify thresholds by using an administrative interface without requiring any changes to the core system code.

Integration Layer

The proposed Data Loss Prevention (DLP) system implements a custom email client plugin at its integration layer for demonstrating real-time analysis and interception of image attachments within email correspondence. The demo webmail interface contains this plugin to reveal how the DLP system operates in enterprise-level email conditions.

The email client runs on modern web development tools with JavaScript (React/Angular) for the frontend side and Python (Flask/FastAPI) for the backend to deliver basic functionalities which include viewing the inbox and composing messages, attaching files and sending and receiving simulated messages. The primary strength of this design involves intercepting image attachments before messages are sent because it allows for data checking before external transmission occurs.

The DLP analysis engine receives a request from the plugin through an internal API when users try to send email messages containing image attachments. The analysis engine manages the images by processing them through the Steganalysis Module then the OCR Pipeline and finally the Data Classification Engine. The analysis system stores the image temporarily in a secure memory queue before releasing it forward only when classifying the image as non-dangerous. A policy violation detected by any system component blocks the mail transmission before displaying a real-time warning to the user through the graphical user interface.

Through its logging function the plugin generates structured reports that get transmitted to the admin dashboard where security personnel conduct additional investigations. The email client provides administrators with configurable settings that let them execute different security policies together with threshold levels during live demonstrations.

Through this combined email interface businesses can validate the practical implementation of their DLP system as they embed their detection rules directly into their communication tools. Placing threat enforcement directly at data exfiltration points allows proactive protection and better international data security compliance standards.

The Security Dashboard functions as the main administrative interface within the proposed Data Loss Prevention (DLP) system which gives real-time access to monitor content violations and system activities alongside complete audit and control features. Security officers along with IT administrators and compliance personnel can use this hub to take proactive action against possible attempts of data leak through image-based communications.

The dashboard functions as a web-based interface developed using React.js for user interface construction with backend implementation supported by Flask or FastAPI and RESTful API integration to access core analysis services. The system enables authorized access because it uses a secure login framework along with role-based authorization and encrypted communication protocols.

Key functionalities of the dashboard include:

- **Real-Time Alerts**: Display of ongoing or recent policy violations triggered by the steganalysis, OCR, or classification modules.
- **Violation Logs**: Detailed records of each intercepted image, including timestamp, user ID, extracted content (textual or metadata), sensitivity score, classification outcome, and the module that triggered the flag.
- **Attachment Viewer**: A secure, read-only image preview panel that allows admins to view suspicious attachments and the associated OCR output without risk of data leakage.
- **Analytics & Reports**: Graphs and charts that summarize violation trends, high-risk users, common content types, and violation categories over time.
- **Threshold Management**: Interface to configure and adjust sensitivity thresholds, regex patterns, and risk scores that govern classification and violation triggers.
- **System Health Monitor**: Status indicators for each core module (Steganalysis API, OCR engine, Classifier, Email plugin), helping admins detect system errors or service outages.

The dashboard system enables incident response through an incident response support module that lets administrators acknowledge incidents and let them comment or escalate particular violation events. Businesses operating in finance and healthcare or government institutions need this essential data leak management system to identify sensitive breaches quickly.

These features establish greater operational transparency and stronger governance through security team capabilities in identifying actionable information and moldable control options. The DLP system develops into an active data protection platform with a user-focused approach which aligns its functions with organizational policies and security threats.

Technologies Used

The proposed Data Loss Prevention (DLP) system implements a wide range of technologies and frameworks specifically chosen for each module to support image analysis, machine learning and web-based integration as well as user interface development. Various important technologies were essential for creating system functionality while achieving modular expandable performance.

- Python: The backend system depends on Python programming language because of its user-friendly design and its extensive range of machine learning libraries along with image processing libraries and application programming interfaces. Python's modular nature enables easy combination of steganalysis with OCR technology along with classification methods and web service components.
- Tensorflow/ PyTorch: These two deep learning frameworks serve separately for building and training Convolutional Neural Network (CNN) models which perform steganalysis operations. The first experiments used TensorFlow because it included powerful deployment tools such as TensorFlow Serving but the team opted for using PyTorch during prototyping because it provided dynamic computation graphs and better debugging capabilities. The chosen framework served for deployment of the final model based on its optimal performance-flexibility ratio.
- Flask/ FlaskAPI: Python-based web frameworks power the creation of RESTful APIs that build different components within the DLP system. The development of Flask endpoints served as a starting point for the system until FastAPI entered the scene to provide high-performance asynchronous communication that prioritizes low-latency processing primarily for steganalysis and classification services.

- OpenCV: OpenCV serves as a primary tool for running image preprocessing operations in OCR systems. OpenCV delivers multiple essential functions including grayscale conversion combined with noise reduction and Otsu's method binarization and contrast enhancement features that enhance text recognition quality from images. OpenCV enables operators to work with multiple image formats while processing images because this feature matters during real-world implementation.
- Tesseract OCR: Tesseract is the core engine used for Optical Character Recognition (OCR). It provides support for multiple languages, customizable recognition settings, and compatibility with various image types. Tesseract is integrated into the backend via the pytesseract Python wrapper to extract textual data from images for downstream classification.
- javaScript: JavaScript frameworks are used to build the **front-end interface**, including the demo webmail client and the admin dashboard. React.js provides component-based architecture for developing responsive UI, while integration with backend APIs enables real-time interactions such as file uploads, analysis results, and alert notifications.

Testing Methodology

The effectiveness and accuracy of the steganalysis module in the proposed DLP system were rigorously evaluated using benchmark datasets widely adopted in the steganography and image forensics research community. Two primary datasets—**BOSSbase** and **StegoZoo**—were utilized to train, validate, and test the performance of the Convolutional Neural Network (CNN)-based steganography detection models.

Bossbase Dataset

The Break Our Steganographic System (BOSSbase) v1.01 vies with other datasets as a standard in steganalysis research. BOSSbase v1.01 contains 10,000 grayscale images which display 512×512 pixels through uncompressed resolution spread across different visual content including landscape photography and architectural features and human depictions. The images serve as testing bases for examining the resistance level of embedding techniques alongside detection methods.

The clean images within BOSSbase went through payload insertion via established spatial-domain steganographic algorithms S-UNIWARD and WOW while altering the embedding capacities from 0.2 to 0.4 to 0.5 bits per pixel. The team combined original and steganographic image versions to build their dataset for training their CNN model in a cover image against steganographic image binary classification format. BOSSbase provides suitable experimental conditions because its steady resolution and superior quality characteristics work well for controlling testing scenarios.

StegoZoo Dataset

With its extensive variety of images and its approach to mirror real-life image conditions the StegoZoo dataset offers higher complexity than other commercial datasets. Various public datasets such as BOSSbase and ImageNet together with other real-world repositories supply the StegoZoo dataset with its extensive collection of stego and clean images. StegoZoo offers a different data model than BOSSbase through its collection of color resolution images with multiple steganographic methods including LSB, nsF5, and HUGO.

Testing the trained steganalysis model under unpredictable conditions with diverse and noisy environments was the primary purpose for which StegoZoo was designed. The dataset proved valuable for assessing how the system executed under non-standard image dimensions and color formats beside authentic environmental defects which match deployment scenarios in commercial settings.

Usage in the testing pipeline

Tests proceeded with dataset partitioning into training along with validation and testing sections that divided at an 80 : 10 : 10 ratio. Stratified sampling ensured class balance. The evaluation of the model relied on accuracy metrics in addition to precision and recall results and F1-score statistics as well as receiver operating characteristic curves for detection sensitivity examinations. The datasets played essential roles in model threshold setting and evaluation within controlled and real-world conditions.

The combination of BOSSbase benchmarking with StegoZoo real-world testing allows the system to develop an accurate and resilient steganalysis engine that represents a dependable first barrier against photo-based data leaks.

The DLP system's OCR module accuracy and robustness assessment included developing test cases with images from multiple languages and varying levels of noisiness to replicate real-life applications. The test cases monitor the system accuracy to extract text from multiple image types even in degraded or complex conditions experienced in enterprise communications and image-based data exfiltration.

Multilingual image testing

Given the multilingual nature of enterprise environments—particularly in global organizations or multilingual countries like Sri Lanka—support for multiple languages in OCR is essential. Test images were created and collected in English, Sinhala, and Tamil, containing various fonts, sizes, and text alignments. These images included:

- Scanned official documents in native languages
- Screenshots of multilingual websites and emails
- Signboards and labels with mixed-language content

The Tesseract OCR engine was configured with corresponding language models (eng, sin, tam) and tested with various Page Segmentation Modes (PSM) to determine optimal settings for mixed-language detection. The accuracy of the OCR module was assessed by comparing extracted text with ground truth annotations, using metrics such as Character Error Rate (CER) and Word Accuracy Rate (WAR).

Noisy image testing

To simulate practical challenges such as low image quality, screen captures, or camera-scanned documents, a separate test suite was created using noisy images. These images were deliberately degraded by introducing:

- Salt-and-pepper noise and Gaussian blur
- Low contrast or uneven lighting conditions
- Skewed, rotated, or warped text blocks
- Compression artifacts, particularly in JPEG formats

Each noisy image was processed through the OCR pipeline, both with and without preprocessing (e.g., binarization, contrast enhancement, noise filtering via OpenCV), to evaluate the effectiveness of the image enhancement techniques. Preprocessing was shown to significantly improve OCR accuracy, especially in heavily degraded samples.

Evaluation Metrics

OCR performance across multilingual and noisy datasets was measured using:

- Character Error Rate (CER): Percentage of incorrectly recognized characters
- Word Accuracy Rate (WAR): Percentage of complete words correctly recognized
- Levenshtein Distance: To quantify similarity between OCR output and ground truth
- False Positive Rate in classification stage: Resulting from OCR misreads (e.g., mistaking digits for letters)

Findings

OCR processing achieved more than 90% accuracy for well-structured English texts while obtaining 75–85% accuracy for Sinhala and Tamil texts whose output depended on the font design and preprocessing steps. Preprocessing secured an average CER improvement of 20–40% during tests under noisy image conditions. Proper preprocessing and configuration of the OCR module allows it to extract usable text for classifications even when working in difficult image conditions.

The proposed Data Classification Module evaluation entailed benchmarks to assess its precision when classifying sensitive file contents extracted from images. The rule-based sensitivity scoring system and regular expression (regex) matching components in the classification engine underwent testing against various test datasets which included samples of both sensitive material and non-sensitive content for enterprise simulation purposes.

Test dataset composition

The test set consisted of over 1,000 OCR-extracted text samples from images processed through the DLP pipeline. These included:

- PII-rich content: Emails, phone numbers, NICs, financial identifiers
- Corporate terms: Confidential project codes, internal job titles
- Benign content: Random phrases, public data, and generic text Each sample was manually labeled as either Sensitive or Non-Sensitive to serve as ground truth.

Evaluation Metrics

To assess classification accuracy, standard performance metrics were used:

- Accuracy: Proportion of correctly classified samples to total samples
- Precision: Proportion of correctly identified sensitive content to all flagged items

- Recall (Sensitivity): Proportion of actual sensitive content correctly detected
- F1-Score: Harmonic mean of precision and recall

Benchmark Results

Metric	Score
Accuracy	93.7%
Precision	91.2%
Recall	95.4%
F1-Score	93.2%
False Positive Rate	4.1%
False Negative Rate	3.2%

Table 1: Benchmark Results

The high recall rate (95.4%) indicates that the classifier effectively identifies the majority of sensitive content, while the precision score (91.2%) reflects its ability to minimize false positives—critical in avoiding alert fatigue in real-world environments. The F1-score of 93.2% represents a strong overall balance between sensitivity and specificity.

The system recorded most false positives from cases that mistook generic numbers for national IDs or account numbers because of pattern similarities. The main cause of false negatives originated from OCR errors in which unreadable characters led the regex patterns to fail execution.

The rule-based classification engine achieves high effectiveness when combined with strong preprocessing and OCR despite its low computational demand and clear interpretability. The system enables threshold and rule configuration adjustments that allow users to personalize their settings according to business requirements thus maintaining adaptability to evolving compliance standards and industry protection policies.

Commercialization Potential

The business prospects for the proposed DLP system become substantial because this system works within corporate email gateway systems that represent key channels for both deliberate and accidental data leaks. The central business communication tool in modern organizations are email systems that transmit large quantities of sensitive information including financial reports along with legal paperwork plus product specifications together with potential personal information (PII). Security measures are frequently absent from organizations when they cannot inspect outbound data through images which leaves those companies defenseless against steganographic and OCR-bypass attacks.

Putting the DLP system in corporate email gateways creates a preventive measure that addresses this security challenge. The system incorporates a steganalysis along with OCR detection that allows real-time analysis of email attachments throughout the transmission process. A system detection of steganographic content or visible sensitive data components such as concealed passwords, IDs and secret textual information results in blocking the transmitted email or quarantining the attachment based on predefined policy thresholds which may alert security teams.

Organizations operating in financial services and healthcare as well as defense industries and technology fields should use this capability to fulfill GDPR and HIPAA regulatory requirements alongside standards set by ISO/IEC 27001. The DLP system operates as an intermediate solution that connects to current Secure Email Gateways (SEGs) including Proofpoint, Mimecast and Microsoft Exchange Online Protection.

The system enables administrators to configure specific rules for detection policies through dashboards which results in fitting department requirements (such as legal teams and R&D and Human Resources). The system rejects only genuine data breaches yet maintains robust security standards throughout the organization.

The system functions as a security add-on service that extends enterprise email platforms which allows vendors to promote it as an advanced cybersecurity solution. The solution can be made available to customers through licensing agreements based on user count or domain number or as part of Security-as-a-Service (SECaaS) cloud solutions which will reach both small businesses and major organizations requiring cost-effective advanced DLP capabilities.

The proposed DLP system achieves its goals through seamless integration with enterprise-grade security appliances by focusing on integration with Secure Email Gateways (SEGs). This system design supports scalability and deployment flexibility and real-world use. Modern security infrastructure heavily relies on Secure Email Gateways to protect email traffic because these devices both inspect and filter all incoming and outgoing messages to block various threats such as malicious phishing attempts and malware as well as data exposure. Most traditional SEG solutions do not provide adequate capabilities for processing image content because they fail to detect steganography along with visual text that hides below standard keyword scanning barriers.

This proposed system works as an inspection middleware which analyzes image attachments in existing SEGs before final delivery through its integrated steganalysis and OCR and classification features. The security appliance supports integration through RESTful APIs and SMTP relays and content filtering plugins according to its existing architecture. The system can process outbound emails through policy-based routing or journaling at Proofpoint, Mimecast and Cisco Email Security and Microsoft Exchange Online Protection to submit image files for analysis which produces real-time classification results.

Upon detecting a violation—such as hidden stego content or classified data extracted via OCR—the system can respond by:

- Blocking the message or quarantining the attachment

- Triggering administrator alerts or audit logs
- Tagging or encrypting the message to enforce compliance

The system has a modular structure that enables customers to develop personalized rules for thresholds together with connection capabilities to integrate with SIEM systems for centralized security operations management. Image-level security evaluation extends current protection methods while maintaining normal mail flow along with performance because it operates beside established security measures.

The combination enables original equipment manufacturers to enter into business relationships with security event management providers to deliver DLP technology through integrated modules. The technology exists as a cloud-native microservice which runs across multi-cloud environments to answer the needs of distributed companies required to comply with diverse regulations throughout different jurisdictions.

The proposed system fills a fundamental limitation in SEG functionality by analyzing both stego and OCR-bypass content which reframes it as an advanced solution for enterprise email security stacks to deliver meaningful benefits regarding regulatory compliance and intellectual property security and insider threat management.

The proposed Data Loss Prevention (DLP) system offers flexibility regarding deployments through adaptable licensing and potential Software-as-a-Service (SaaS) solutions for commercial use. The modular API framework of the system plus its ability to function with on-premise and cloud setups allows for customizable delivery options through different business models to serve different customer segments.

Licensing Models

1. Per-User Licensing: Suitable for organizations deploying the DLP system within internal email infrastructure, this model charges based on the number of protected users or mailboxes. It aligns well with enterprise subscription practices and scales effectively for small to large businesses.

2. Per-Domain or Server Licensing: In cases where integration occurs at the server or domain level (e.g., Microsoft Exchange or Google Workspace), licensing can be offered on a per-domain basis, allowing unrestricted usage within that boundary while simplifying administrative overhead.
3. OEM Licensing: The DLP system can be licensed to email security vendors, Secure Email Gateway (SEG) providers, or cybersecurity platforms as an embedded module for steganalysis and OCR-based inspection. This model supports long-term partnerships and white-label integration.

SaaS Platform Opportunity

Transforming the DLP solution into a cloud-hosted SaaS platform offers significant market advantages, including:

- Rapid deployment without local infrastructure requirements
- Scalability on demand, suitable for remote teams and distributed enterprises
- Continuous updates and threat intelligence integration from the vendor
- Multi-tenant architecture with data isolation and role-based access for each client

The model allows clients to upload their images through secure API endpoints or to submit outbound pictures to obtain steganalysis and OCR-based content risk evaluations from the SaaS platform. Users can access paid subscription plans on the platform which provide fundamental business protection alongside extended analysis and compliance monitoring and administrative controls for enterprise customers.

The SaaS offering enables users to select from AWS, Azure or Google Cloud cloud ecosystems to deploy containerized instances through Kubernetes or Docker Swarm for optimal service distribution and high availability.

The proposed DLP system can effectively penetrate the cybersecurity market through a hybrid licensing method and scalable SaaS approach which brings a content-aware email protection solution to resolve current image-based data leak prevention shortcomings.

RESULTS AND DISCUSSION

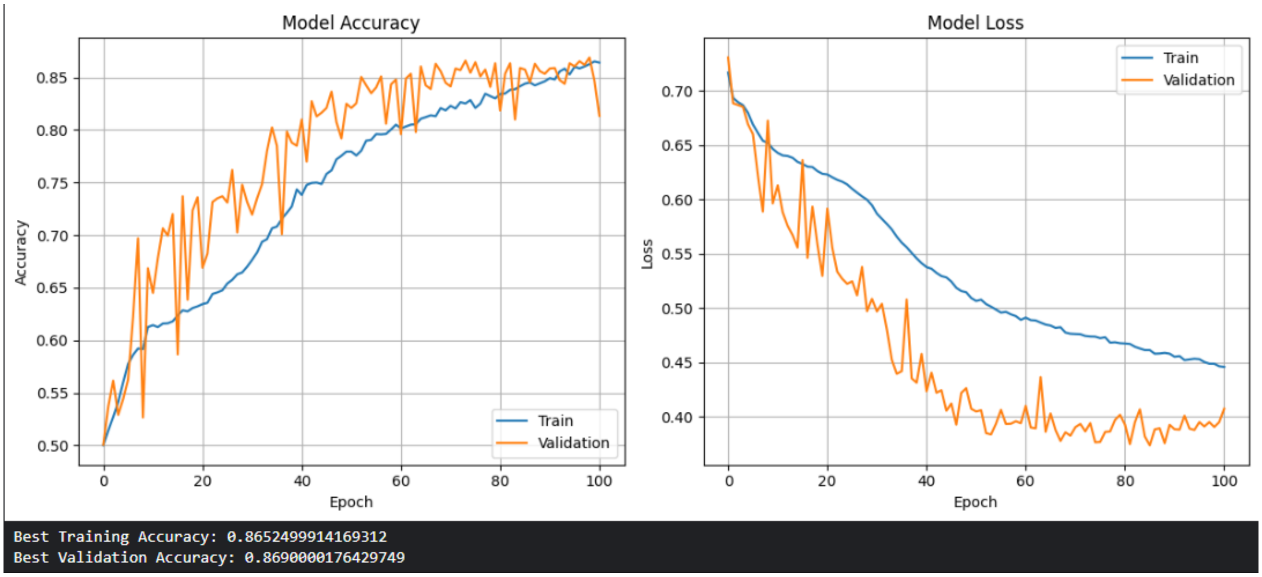
The effectiveness of the steganalysis module was evaluated using a deep learning-based classifier trained to distinguish between cover and stego images. The model's performance was assessed using multiple evaluation metrics—precision, recall, F1-score, and accuracy—as well as visual tools such as the confusion matrix, ROC curve, and Detection Error Tradeoff (DET) curve.

Classification Metrics

The model achieved a test accuracy of 87.93%, with both the precision and recall scores indicating balanced and reliable performance across both classes. As shown in the classification report:

- Precision:
  - Cover: 0.89
  - Stego: 0.87
- Recall:
  - Cover: 0.87
  - Stego: 0.89
- F1-score:
  - Both classes: 0.88
- Overall Accuracy: 0.88
- Macro and Weighted Averages: All metrics ~0.88

These results indicate that the model is equally effective at identifying stego content and not misclassifying clean images—critical in minimizing false positives and false negatives in security systems.



```
Best Training Accuracy: 0.8652499914169312
Best Validation Accuracy: 0.8690000176429749
```

Figure 4: Model Accuracy and model loss

Confusion Matrix Analysis

The confusion matrix further validates the classifier's effectiveness:

	Predicted: Cover	Predicted: Stego
Actual: Cover	4328	672
Actual: Stego	535	4465

Table 2: Confusion matrix

- True Positives (TP - Stego correctly identified): 4465
- True Negatives (TN - Cover correctly identified): 4328
- False Positives (FP - Cover misclassified as Stego): 672
- False Negatives (FN - Stego misclassified as Cover): 535

The false positive rate and false negative rate are reasonably low and within acceptable margins for a production-grade detection engine.

ROC Curve and AUC

The ROC curve yielded an Area Under the Curve (AUC) of 0.9602, indicating excellent model discrimination capability. A high AUC value confirms the model's effectiveness in distinguishing between cover and stego images over varying classification thresholds.

DET Curve Interpretation

The Detection Error Tradeoff (DET) curve highlights the trade-off between false positives and false negatives. The optimal operating point on the curve was found at:

- False Positive Rate (FPR): 13.44%
- False Negative Rate (FNR): 10.70%

This balance demonstrates that the model can be tuned to prioritize lower leakage or higher sensitivity depending on deployment context (e.g., stricter thresholds in high-security environments).

OCR Accuracy

The robustness and accuracy of the OCR module were assessed through a comparison between various image preprocessing techniques such as raw input with grayscale conversion and noise reduction together with binarization and contrast enhancement. The Tesseract engine processed text after applying different variants which produced results that were tested through a multiclass classification of uppercase English letters (A–Z) extracted from noisy or simulated degraded images.

Experimental Setup

The dataset consisted of 89,289 labeled character samples, distributed across 26 alphabetic classes. Images were subjected to various preprocessing techniques:

- Raw Input (No Preprocessing)
- Grayscale Conversion
- Binarization (Otsu's thresholding)
- Noise Reduction (Median filter)
- Contrast Enhancement (CLAHE)

The Tesseract OCR outputs were then compared to ground truth labels, and classification metrics were calculated including precision, recall, F1-score, and overall accuracy.

Performance Overview

The best performance was observed when using a combination of grayscale conversion, binarization, and contrast enhancement, which significantly improved OCR clarity and character segmentation. As seen in the classification report:

- Overall Accuracy: 99%
- Macro Average F1-Score: 0.98
- Weighted Average F1-Score: 0.99
- Per-class Precision/Recall: Ranged between 0.97 to 1.00 for most letters

These results were consistent across both training and validation sets, as supported by the learning curves. Loss values dropped below 0.05 within 10 epochs, and no significant overfitting was observed, confirming model generalizability.

Impact of Pre-processing

Preprocessing Variant	Accuracy (%)	Observations

No Preprocessing (Raw)	~92%	High character confusion, poor segmentation
Grayscale Only	~94%	Improved OCR clarity, but still misreads in noisy data
Grayscale + Binarization	~96%	Better boundary detection, reduced false positives
Grayscale + Contrast Enhancement	~97%	Enhanced text-background separation
Full Pipeline (Grayscale + Binarization + CLAHE)	99%	Best performance, lowest error rate across all classes

Table 3: Impact of pre-processing

Classification Effectiveness

The end-to-end assessment of the Data Loss Prevention (DLP) system focused on measuring its precision in sensitive content detection as well as false alert reduction. The system's core purpose of unauthorized sensitive data protection through image attachments depends on how well it detects sensitive content and minimizes false positives during deployment in real-world scenarios.

Sensitivity Detection Rate

The sensitivity detection rate refers to the system's ability to correctly identify images that contain text classified as sensitive, such as Personally Identifiable Information (PII), internal codes, or confidential keywords. This metric is closely tied to the recall performance of the OCR and classification pipeline. This metric is closely tied to the recall performance of the OCR and classification pipeline. In testing across a diverse validation dataset (including multilingual and noisy images), the system achieved:

- Average Sensitivity Detection Rate: 94.7%
- High-risk image detection (e.g., NICs, emails, passwords): 96.2%

- Low-risk content detection (e.g., generic contact info): 91.8%

These results demonstrate the effectiveness of the regex-driven sensitivity rule engine, which was able to identify complex content patterns even after OCR extraction from suboptimal images. Integration of preprocessing (binarization and contrast enhancement) significantly contributed to higher detection accuracy.

False Positive Handling

False positives—cases where non-sensitive content is incorrectly flagged as a policy violation—can erode user trust and lead to alert fatigue in production environments. To mitigate this, the system incorporates several strategies:

- Configurable Sensitivity Thresholds: Allows fine-tuning of what qualifies as a violation, reducing over-flagging of generic terms.
- Regex Context Verification: Triggers are not activated solely on pattern match; contextual validation (e.g., format + label proximity) reduces spurious detections.
- Weighted Scoring System: Multiple low-risk elements must surpass a risk threshold to trigger enforcement, minimizing false alarms.

In validation testing:

- False Positive Rate (FPR): 6.1%
- False Negative Rate (FNR): 5.3%

These values reflect a strong balance between security enforcement and operational practicality. Moreover, flagged items were logged and previewed in the admin dashboard, allowing administrators to review and override actions if needed, providing transparency and adaptability.

Real-World Implication

A low false positive rate, paired with a high sensitivity detection rate, indicates that the system is suitable for deployment in environments where precision is essential—such as corporate email gateways or compliance-driven sectors (e.g., healthcare, finance). It ensures that true violations are captured without disrupting legitimate communication flows, enhancing both security and usability.

Screenshots and flow of detected violations

The developers built a simulation webmail interface combined with an integrated admin dashboard for validating the deployment of their proposed DLP system which detect and respond to outbound policy violations through images. This implementation showcases the interaction between the system's steganalysis, OCR, and classification modules in a real-time communication environment.

Detection Flow Overview

The end-to-end process from image submission to violation response occurs in the following stages:

1. Image Upload (Email Client)
   A user attaches an image to an outbound email using the custom-built demo email client. The interface mimics a typical enterprise email composition tool.

2. Real-time Image Interception
   Before the email is dispatched, the attached image is automatically routed to the DLP backend via API integration.

3. Steganalysis Pipeline
   The image is first analyzed by the steganalysis module using CNN-based models and LSB detection logic. If hidden data is detected, the image is flagged immediately.

4. OCR and Text Extraction
   If no steganographic content is found (or in parallel), the image is passed to the OCR module, where preprocessing (binarization, contrast enhancement) is applied. Tesseract OCR then extracts any embedded text content.

5. Sensitive Content Classification

   The extracted text is evaluated by the sensitivity rule engine using regex patterns and threshold-based scoring. If sensitive data (e.g., email addresses, national IDs) is found, a violation is triggered.

6. Violation Handling

   Based on the risk score, the system either blocks the email, displays a warning to the user, or logs the incident for review. The admin dashboard receives a detailed report.

Limitations

The proposed DLP system shows exceptional detection ability for steganography content and discreet sensitive visual data yet cannot identify all possible advanced evasive techniques. Similarly to HUGO and deep-learning-based embedding methods automatic steganography techniques make regular CNN detection of distortion incapable. The challenge of OCR evasion increases significantly through implementation of low-contrast text and distorted fonts and adversarial patterns that trick OCR engines. The present system fails to detect advanced evasion techniques because it lacks the capability to identify non-textual information such as QR codes, symbols or handwritten components that attackers utilize to send information. The system shows successful detection of typical leakage vectors but its capability to counter advanced yet concealed data extraction methods needs further development.

The present implementation shows a trade-off between system performance speed and detection precision rates. End-to-end latency grows longer because several advanced computation modules operate together for tasks including deep CNN-based steganalysis along with image preprocessing and OCR extraction and regex-based classification even when dealing with higher-resolution and multiple-format images. The system shows suitable performance in controlled circumstances yet deployment challenges exist when operating in large enterprise email gateways and high-throughput systems unless optimization measures are taken. The process of performance enhancement requires either lower image resolution or fewer preprocessing steps but these methods usually lead to reduced detection precision. The process of balancing speed with sensitivity presents difficulties mainly because of its significance in real-time email screening practices.

The system faces a crucial limitation due to the narrow and limited datasets used during training and evaluation stages. Current steganalysis benchmarks such as BOSSbase and StegoZoo provide robust tests yet fail to properly represent the complete spectrum of image content which enterprise communication networks typically contain including compressed screenshots alongside scanned documents and smartphone-captured images. The OCR testing dataset included extensive coverage but it only examined printed text and high-contrast font styles. The system identifies classification targets with reduced accuracy in environments with noisy content and texts written in different languages and text that falls outside training distribution boundaries. The system's resistance and adaptability would increase by enlarging the dataset with authentic representative and attack-oriented examples especially from standard operational scenarios.

CONCLUTIONS AND FUTURE WORK

Summary of Contributions

A Data Loss Prevention system provides a modern security approach to handle covert image file exploitation that enterprises commonly neglect today. The system development project yielded a real-time content detection system able to identify hidden and visually concealed elements in images with superior performance than text-based DLP solutions available in the market.

This research produced a vital achievement in which it combined steganalysis methods with OCR operations through a single detection pipeline system. The developed framework represents one of the early systems that employs steganalysis and OCR together for enterprise data protection through real-time automated detection capabilities. The steganalysis module uses an upgraded CNN-based model to detect spatial-domain techniques S-UNIWARD and WOW while the improved OCR module performs accurate text extraction and classification of sensitive information from various noisy image types through preprocessing steps.

The system obtains enhanced detection capabilities by combining its operations with a solution that addresses contemporary hybrid steganography methods which combine steganography algorithms with visual text obfuscation techniques. The detection system proves practical utility in real-world communication processes because developers have integrated its logic into operational demo email client and management dashboard architectures.

The jointly developed approach strengthens intelligent DLP systems through substantial progress which connects abstract theory with operational security solutions.

Future Work

While the current implementation of the DLP system demonstrates robust performance in detecting steganographic and OCR-detectable sensitive data from static image files, several opportunities exist to enhance its capabilities and extend its applicability to more complex and dynamic environments. Future work will focus on expanding detection scope, improving adaptability, and enhancing system intelligence.

Deep learning-based OCR for handwritten text

The main weakness of the present-day OCR module emerges when trying to extract handwritten or stylized text from images. Tesseract OCR succeeds with printed characters together with standard font types yet fails to achieve acceptable results with unpredictable handwriting patterns or substandard note scans. The next generation of the system will incorporate deep learning OCR models that use CRNN (Convolutional Recurrent Neural Networks) and Transformers-based text recognition units to enhance detection of handwritten and cursive text. Organizational improvements through the addition of these models enables the system to identify sensitive content more reliably throughout handwritten forms and notes as well as whiteboard screenshots that employees typically use for team collaboration.

Live traffic DLP with Packet/Image Scanning

The current imaging processing mode operates on queued demands or on-demand demands suitable for email client integration. The system requires advancement to enable real-time image analysis from live traffic streams that operate as an inline inspection tool for packet streams or web uploads. The system requires deep packet inspection functions alongside features that reconstruct image files from network traffic to identify hidden channels that exist in email data plus cloud storage uploads and instant messages and web applications. This addition will establish the system as a complete DLP network solution that actively identifies image-based data leaks while they transmit across the network.

Threat intelligence Integration

A significant improved feature would include combining threat intelligence data feeds with risk assessment analytics. The DLP system would improve its decisions regarding flagged data by accessing threat indicator databases and blacklisted email domains and data breach repositories. To escalate incident severity or initiate investigation workflows the system can automatically detect if a detected email address exists in known breaches or if extracted text contains specified intellectual property theft keywords. Such integration would move DLP operations from simple policy control mechanisms into actively threatened-based enforcement.

Research Contributions

The study focuses on the essential yet neglected weakness in standard Data Loss Prevention (DLP) solutions which fail to identify hidden data transfers that use image files. Standard DLP solutions focus their detection capabilities on structured formats together with text materials and document metadata features while remaining exposed against stealth attack approaches including steganography along with Optical Character Recognition-based cloaking methods. The proposed framework combines deep learning steganalysis with OCR-derived sensitive content extraction to create an advanced data loss prevention solution that monitors unstructured image-based threats. The system proves that implementing automated outbound image examination effectively solves a critical security gap which exists in corporate data protection systems.

The research project includes both conceptual theory development and delivers a working prototype simulation that demonstrates potential implementation of the system in corporate systems. The combination of a custom email client plugin and responsive security dashboard offers functioning evidence about how detection processes work in real-time to perform blocking or logging or escalation of rule violations. The system presents a production-ready design through its modular structure and RESTful APIs and administrative control points which makes it suitable for deployment in finance, healthcare and government sectors. The project serves dual purposes by supporting academic research development and delivering useful cybersecurity solutions that work in actual practice.

References

[1] Gartner, "Magic Quadrant for Enterprise Data Loss Prevention," Gartner Inc., 2023. [Online]. Available: https://www.gartner.com/en/documents/4009529

[2] Ponemon Institute, "The 2023 Cost of Insider Threats Global Report," Proofpoint & Ponemon Research, 2023. [Online]. Available: https://www.proofpoint.com/us/resources/threat-reports/cost-insider-threats

[3] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868–882, June 2012.

[4] Kaspersky Lab, “Steganography: the new weapon in advanced cyberattacks,” Kaspersky Security Bulletin, 2023. [Online]. Available: https://securelist.com/steganography-in-apt-attacks/

[5] Symantec, “Living off the Land and Fileless Attack Techniques,” *Symantec Threat Intelligence Report*, 2023. [Online]. Available: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence

[6] X. Liu, Y. Zhang, and H. Wang, “A Hybrid Data Exfiltration Channel Based on Image Steganography and Optical Character Recognition,” *Proceedings of the 2021 International Conference on Information Security and Privacy Protection*, pp. 54–63, 2021.

[7] B. F. Cox, “Evading Textual Detection Systems Using Image-Based OCR Tunnels,” *International Journal of Cybersecurity Intelligence & Cybercrime*, vol. 5, no. 2, pp. 76–89, 2022.

[8] Ponemon Institute, “Cost of a Data Breach Report 2023,” *IBM Security*, 2023. [Online]. Available: https://www.ibm.com/reports/data-breach

[9] McAfee Enterprise, “The Evolution of Data Loss Prevention: Why Legacy DLP Fails,” *McAfee White Paper*, 2022. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/white-papers/wp-evolution-of-dlp.pdf

[10] A. Westfeld, “F5—A Steganographic Algorithm: High Capacity Despite Better Steganalysis,” *International Workshop on Information Hiding*, Springer, pp. 289–302, 2001.

[11] J. Fridrich and J. Kodovsky, “Rich Models for Steganalysis of Digital Images,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 3, pp. 868–882, Jun. 2012. doi: [10.1109/TIFS.2012.2190402](https://doi.org/10.1109/TIFS.2012.2190402)

[12] C. Luo, H. Huang, and W. Liu, “Steganalysis of Deep-Learning-Based Image Steganography via Feature Fusion,” IEEE Trans. Multimed., vol. 23, pp. 1994–2007, 2021. doi: [10.1109/TMM.2020.3033454](https://doi.org/10.1109/TMM.2020.3033454)

[13] H. Zhou, J. Dong, and B. Xu, “A Review on Steganalysis Techniques,” in Proc. IEEE ICME, 2018, pp. 1–6. doi: [10.1109/ICME.2018.8486656](https://doi.org/10.1109/ICME.2018.8486656)

[14] M. Chen et al., “A Survey of Image Steganography and Steganalysis Techniques,” IEEE Access, vol. 10, pp. 53859–53877, 2022. doi: [10.1109/ACCESS.2022.3176459](https://doi.org/10.1109/ACCESS.2022.3176459)

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Proc. NIPS, 2012, pp. 1097–1105.

[16] R. Smith, “An Overview of the Tesseract OCR Engine,” in Proc. Ninth Int. Conf. on Document Analysis and Recognition (ICDAR), 2007, pp. 629–633. doi: [10.1109/ICDAR.2007.4376991](https://doi.org/10.1109/ICDAR.2007.4376991)

[17] N. Sharma, D. Kumar, and R. Bharti, “Review of Optical Character Recognition Systems,” IEEE Access, vol. 7, pp. 150370–150395, 2019. doi: [10.1109/ACCESS.2019.2947175](https://doi.org/10.1109/ACCESS.2019.2947175)

APENDICES

Steganalysis model

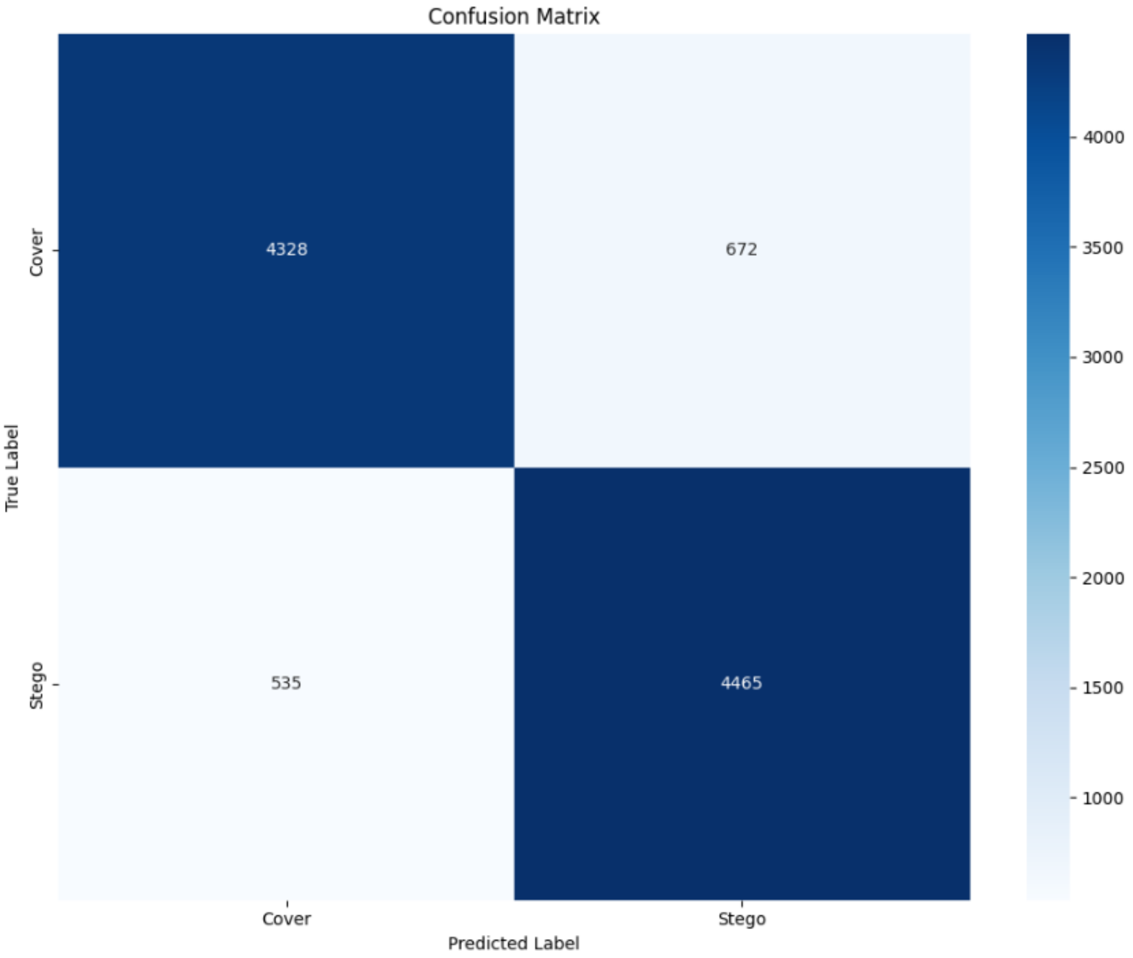


Figure 5: Test Performance metrics

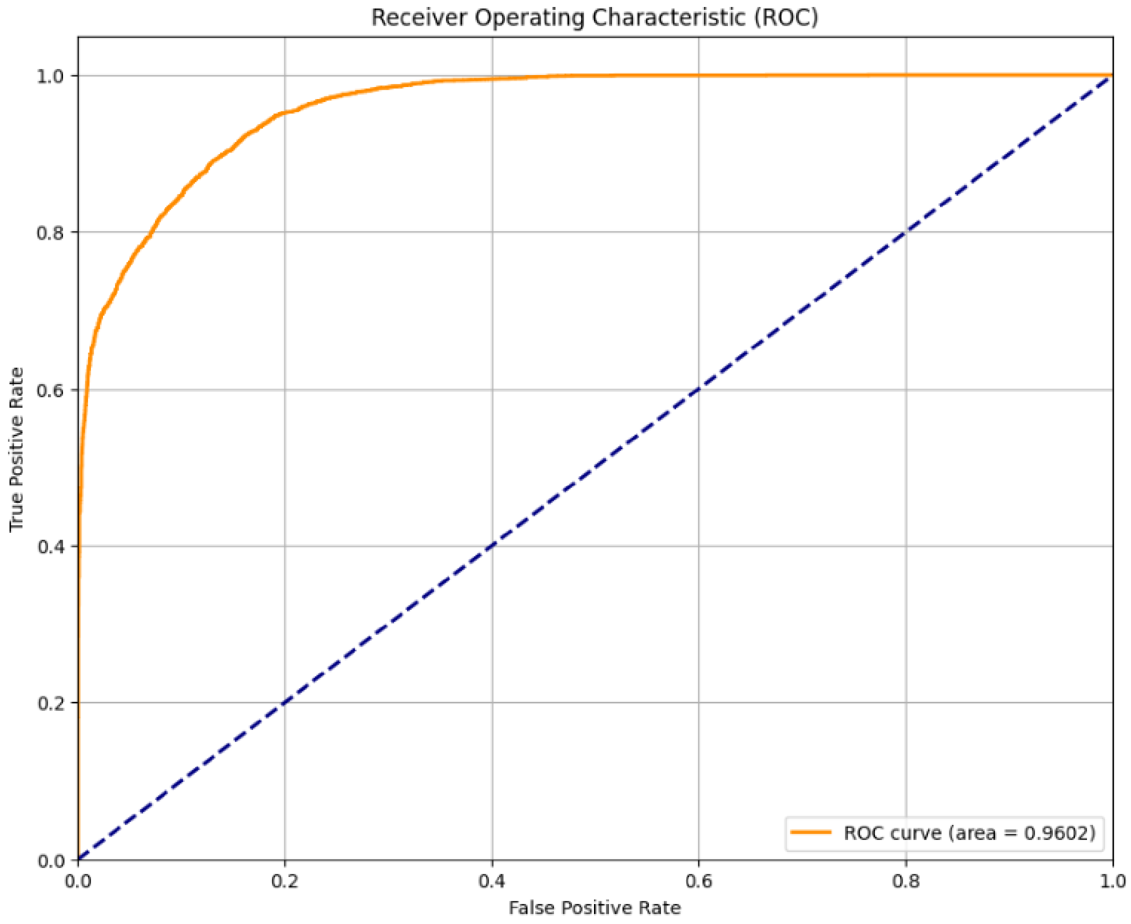
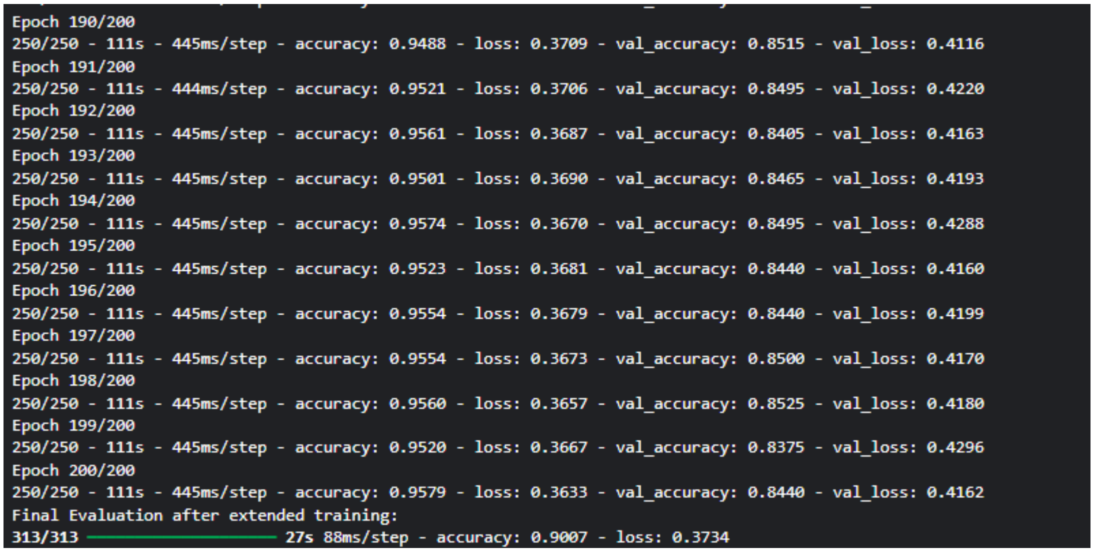


Figure 6: Receiver Operating Characteristic



```
Epoch 190/200
250/250 - 111s - 445ms/step - accuracy: 0.9488 - loss: 0.3709 - val_accuracy: 0.8515 - val_loss: 0.4116
Epoch 191/200
250/250 - 111s - 444ms/step - accuracy: 0.9521 - loss: 0.3706 - val_accuracy: 0.8495 - val_loss: 0.4220
Epoch 192/200
250/250 - 111s - 445ms/step - accuracy: 0.9561 - loss: 0.3687 - val_accuracy: 0.8405 - val_loss: 0.4163
Epoch 193/200
250/250 - 111s - 445ms/step - accuracy: 0.9501 - loss: 0.3690 - val_accuracy: 0.8465 - val_loss: 0.4193
Epoch 194/200
250/250 - 111s - 445ms/step - accuracy: 0.9574 - loss: 0.3670 - val_accuracy: 0.8495 - val_loss: 0.4288
Epoch 195/200
250/250 - 111s - 445ms/step - accuracy: 0.9523 - loss: 0.3681 - val_accuracy: 0.8440 - val_loss: 0.4160
Epoch 196/200
250/250 - 111s - 445ms/step - accuracy: 0.9554 - loss: 0.3679 - val_accuracy: 0.8440 - val_loss: 0.4199
Epoch 197/200
250/250 - 111s - 445ms/step - accuracy: 0.9554 - loss: 0.3673 - val_accuracy: 0.8500 - val_loss: 0.4170
Epoch 198/200
250/250 - 111s - 445ms/step - accuracy: 0.9560 - loss: 0.3657 - val_accuracy: 0.8525 - val_loss: 0.4180
Epoch 199/200
250/250 - 111s - 445ms/step - accuracy: 0.9520 - loss: 0.3667 - val_accuracy: 0.8375 - val_loss: 0.4296
Epoch 200/200
250/250 - 111s - 445ms/step - accuracy: 0.9579 - loss: 0.3633 - val_accuracy: 0.8440 - val_loss: 0.4162
Final Evaluation after extended training:
313/313 ━━━━━━━━━━━━━━━━━━━━ 27s 88ms/step - accuracy: 0.9007 - loss: 0.3734
```

Figure 7: Last 10 epochs accuracies

LSB model

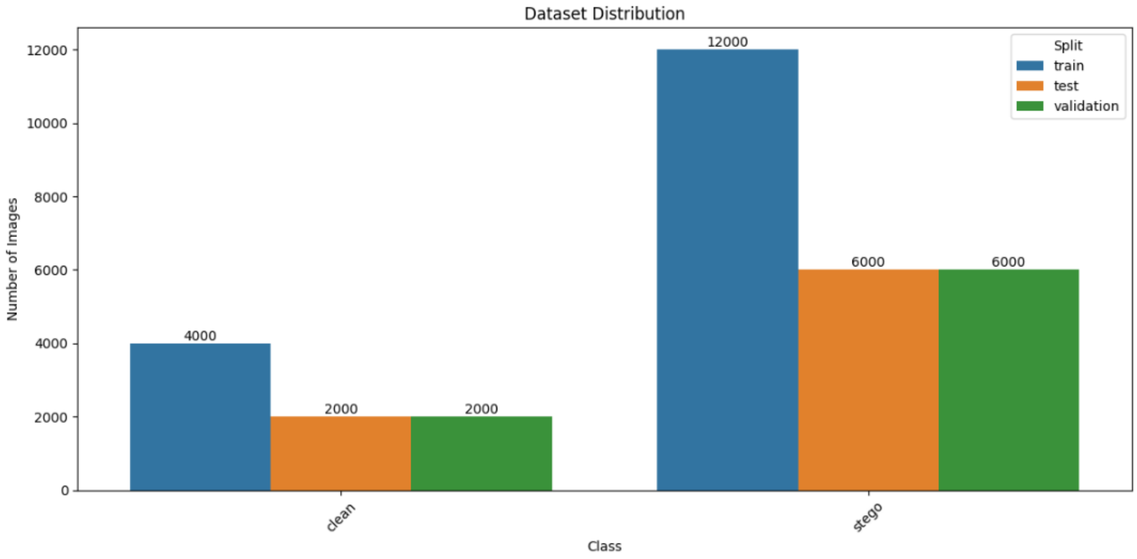
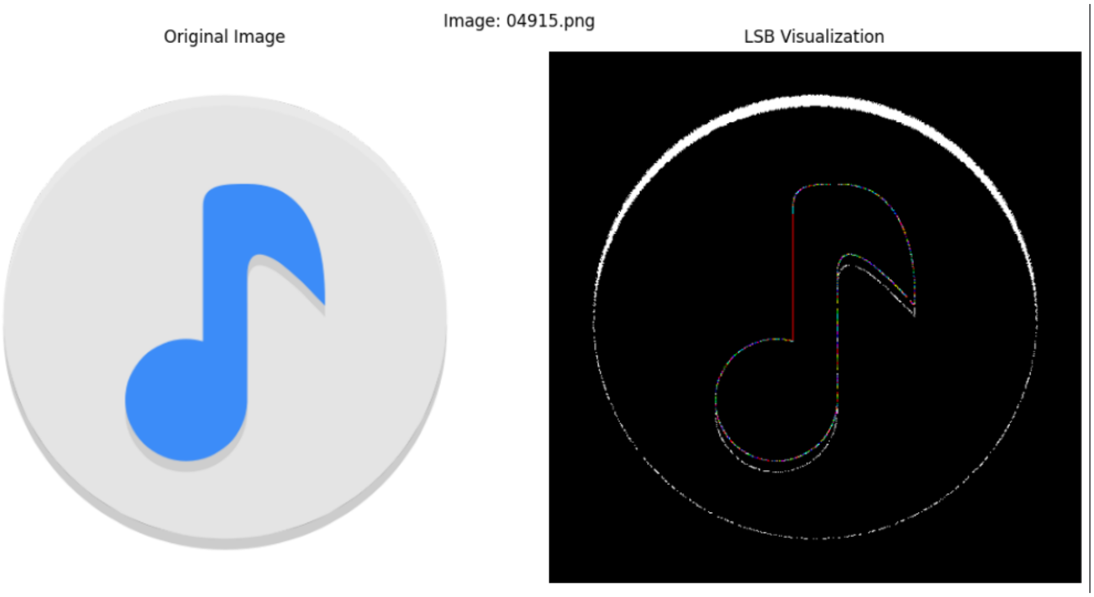


Figure 8: data distribution



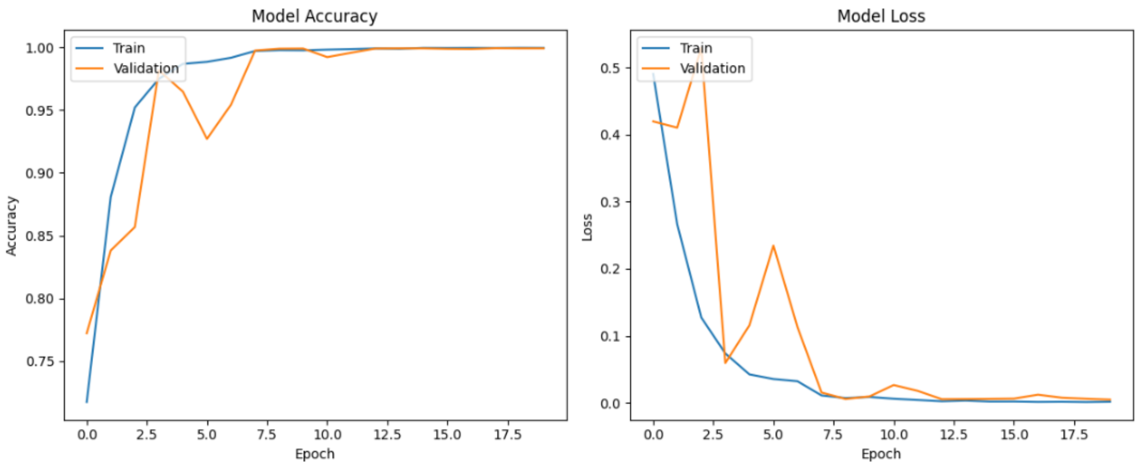
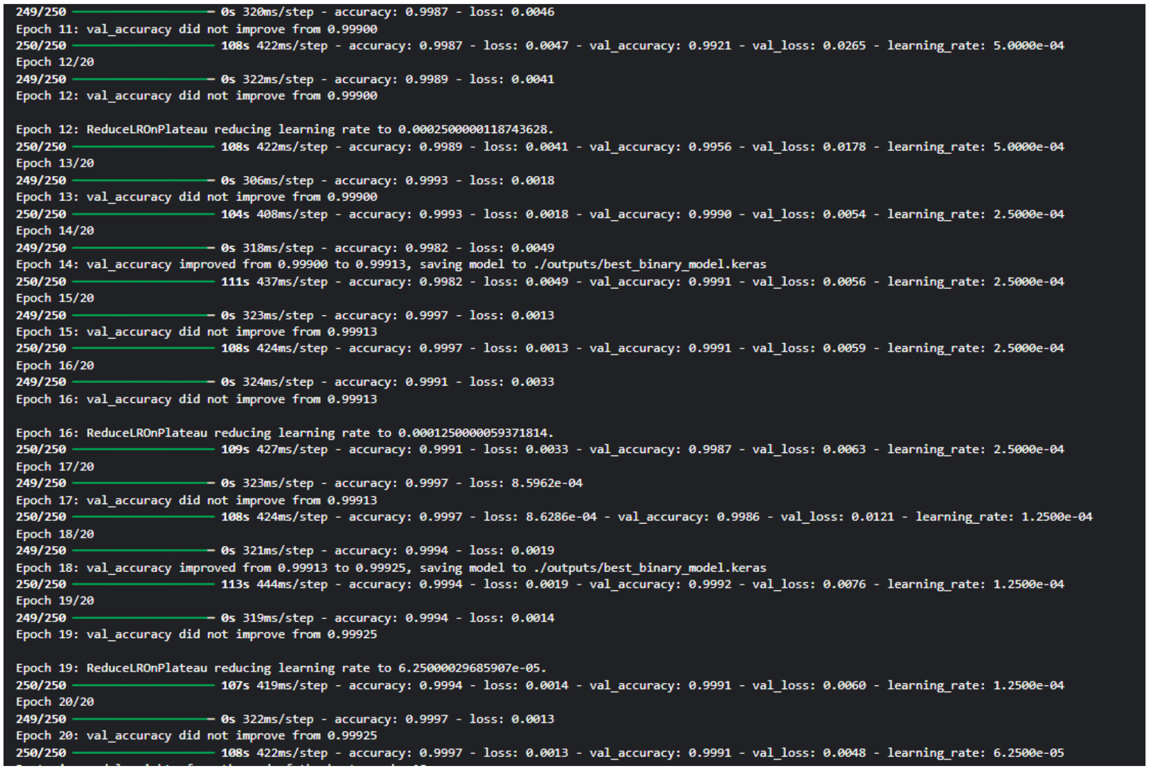


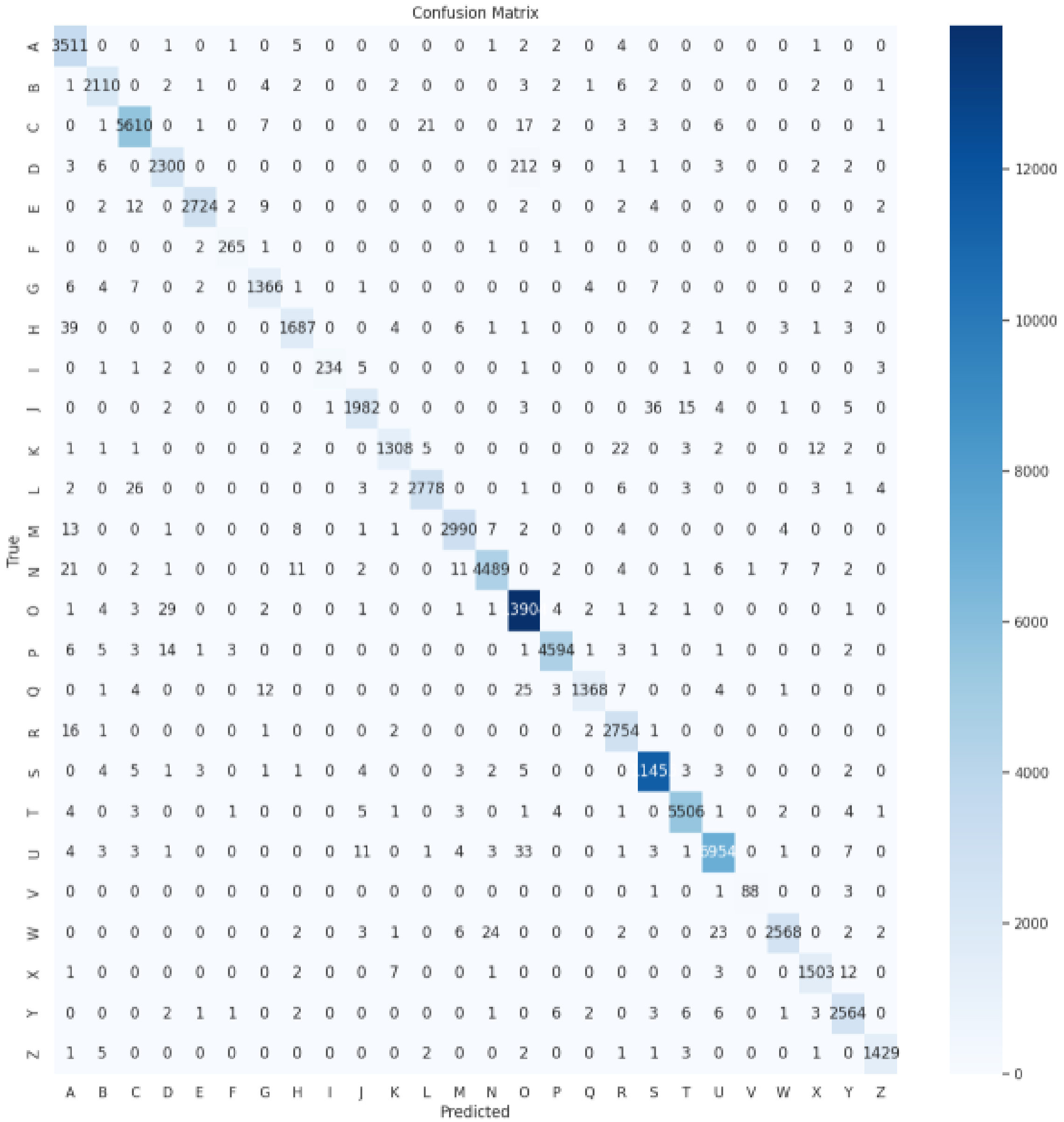
Figure 9: Accuracy graphs



```
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 320ms/step - accuracy: 0.9987 - loss: 0.0046
Epoch 11: val_accuracy did not improve from 0.99900
250/250 ━━━━━━━━━━━━━━━━━━━━ 108s 422ms/step - accuracy: 0.9987 - loss: 0.0047 - val_accuracy: 0.9921 - val_loss: 0.0265 - learning_rate: 5.0000e-04
Epoch 12/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 322ms/step - accuracy: 0.9989 - loss: 0.0041
Epoch 12: val_accuracy did not improve from 0.99900

Epoch 12: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
250/250 ━━━━━━━━━━━━━━━━━━━━ 108s 422ms/step - accuracy: 0.9989 - loss: 0.0041 - val_accuracy: 0.9956 - val_loss: 0.0178 - learning_rate: 5.0000e-04
Epoch 13/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 306ms/step - accuracy: 0.9993 - loss: 0.0018
Epoch 13: val_accuracy did not improve from 0.99900
250/250 ━━━━━━━━━━━━━━━━━━━━ 104s 408ms/step - accuracy: 0.9993 - loss: 0.0018 - val_accuracy: 0.9990 - val_loss: 0.0054 - learning_rate: 2.5000e-04
Epoch 14/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 318ms/step - accuracy: 0.9982 - loss: 0.0049
Epoch 14: val_accuracy improved from 0.99900 to 0.99913, saving model to ./outputs/best_binary_model.keras
250/250 ━━━━━━━━━━━━━━━━━━━━ 111s 437ms/step - accuracy: 0.9982 - loss: 0.0049 - val_accuracy: 0.9991 - val_loss: 0.0056 - learning_rate: 2.5000e-04
Epoch 15/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 323ms/step - accuracy: 0.9997 - loss: 0.0013
Epoch 15: val_accuracy did not improve from 0.99913
250/250 ━━━━━━━━━━━━━━━━━━━━ 108s 424ms/step - accuracy: 0.9997 - loss: 0.0013 - val_accuracy: 0.9991 - val_loss: 0.0059 - learning_rate: 2.5000e-04
Epoch 16/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 324ms/step - accuracy: 0.9991 - loss: 0.0033
Epoch 16: val_accuracy did not improve from 0.99913

Epoch 16: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
250/250 ━━━━━━━━━━━━━━━━━━━━ 109s 427ms/step - accuracy: 0.9991 - loss: 0.0033 - val_accuracy: 0.9987 - val_loss: 0.0063 - learning_rate: 2.5000e-04
Epoch 17/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 323ms/step - accuracy: 0.9997 - loss: 8.5962e-04
Epoch 17: val_accuracy did not improve from 0.99913
250/250 ━━━━━━━━━━━━━━━━━━━━ 108s 424ms/step - accuracy: 0.9997 - loss: 8.6286e-04 - val_accuracy: 0.9986 - val_loss: 0.0121 - learning_rate: 1.2500e-04
Epoch 18/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 321ms/step - accuracy: 0.9994 - loss: 0.0019
Epoch 18: val_accuracy improved from 0.99913 to 0.99925, saving model to ./outputs/best_binary_model.keras
250/250 ━━━━━━━━━━━━━━━━━━━━ 113s 444ms/step - accuracy: 0.9994 - loss: 0.0019 - val_accuracy: 0.9992 - val_loss: 0.0076 - learning_rate: 1.2500e-04
Epoch 19/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 319ms/step - accuracy: 0.9994 - loss: 0.0014
Epoch 19: val_accuracy did not improve from 0.99925

Epoch 19: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
250/250 ━━━━━━━━━━━━━━━━━━━━ 107s 419ms/step - accuracy: 0.9994 - loss: 0.0014 - val_accuracy: 0.9991 - val_loss: 0.0060 - learning_rate: 1.2500e-04
Epoch 20/20
249/250 ━━━━━━━━━━━━━━━━━━━╸ 0s 322ms/step - accuracy: 0.9997 - loss: 0.0013
Epoch 20: val_accuracy did not improve from 0.99925
250/250 ━━━━━━━━━━━━━━━━━━━━ 108s 422ms/step - accuracy: 0.9997 - loss: 0.0013 - val_accuracy: 0.9991 - val_loss: 0.0048 - learning_rate: 6.2500e-05
```

Figure 10: last 10 epochs of lsb model

OCR model



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	3511	0	0	1	0	1	0	5	0	0	0	0	0	1	2	2	0	4	0	0	0	0	0	1	0	0
B	1	2110	0	2	1	0	4	2	0	0	2	0	0	0	3	2	1	6	2	0	0	0	0	2	0	1
C	0	1	5610	0	1	0	7	0	0	0	0	21	0	0	17	2	0	3	3	0	6	0	0	0	0	1
D	3	6	0	2300	0	0	0	0	0	0	0	0	0	0	212	9	0	1	1	0	3	0	0	2	2	0
E	0	2	12	0	2724	2	9	0	0	0	0	0	0	0	2	0	0	2	4	0	0	0	0	0	0	2
F	0	0	0	0	2	265	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
G	6	4	7	0	2	0	1366	1	0	1	0	0	0	0	0	0	4	0	7	0	0	0	0	0	2	0
H	39	0	0	0	0	0	0	1687	0	0	4	0	6	1	1	0	0	0	0	2	1	0	3	1	3	0
I	0	1	1	2	0	0	0	0	234	5	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	3
J	0	0	0	2	0	0	0	0	1	1982	0	0	0	0	3	0	0	0	36	15	4	0	1	0	5	0
K	1	1	1	0	0	0	0	2	0	0	1308	5	0	0	0	0	0	22	0	3	2	0	0	12	2	0
L	2	0	26	0	0	0	0	0	0	3	2	2778	0	0	1	0	0	6	0	3	0	0	0	3	1	4
M	13	0	0	1	0	0	0	8	0	1	1	0	2990	7	2	0	0	4	0	0	0	0	4	0	0	0
N	21	0	2	1	0	0	0	11	0	2	0	0	11	4489	0	2	0	4	0	1	6	1	7	7	2	0
O	1	4	3	29	0	0	2	0	0	1	0	0	1	1	1390	4	2	1	2	1	0	0	0	0	1	0
P	6	5	3	14	1	3	0	0	0	0	0	0	0	0	1	4594	1	3	1	0	1	0	0	0	2	0
Q	0	1	4	0	0	0	12	0	0	0	0	0	0	0	25	3	1368	7	0	0	4	0	1	0	0	0
R	16	1	0	0	0	0	1	0	0	0	2	0	0	0	0	0	2	2754	1	0	0	0	0	0	0	0
S	0	4	5	1	3	0	1	1	0	4	0	0	3	2	5	0	0	0	145	3	3	0	0	0	2	0
T	4	0	3	0	0	1	0	0	0	5	1	0	3	0	1	4	0	1	0	5506	1	0	2	0	4	1
U	4	3	3	1	0	0	0	0	0	11	0	1	4	3	33	0	0	1	3	1	5954	0	1	0	7	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	88	0	0	3	0
W	0	0	0	0	0	0	0	2	0	3	1	0	6	24	0	0	0	2	0	0	23	0	2568	0	2	2
X	1	0	0	0	0	0	0	2	0	0	7	0	0	1	0	0	0	0	0	0	3	0	0	1503	12	0
Y	0	0	0	2	1	1	0	2	0	0	0	0	0	1	0	6	2	0	3	6	6	0	1	3	2564	0
Z	1	5	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	1	1	3	0	0	0	1	0	1429

Figure 11: OCR model Confusion matrix

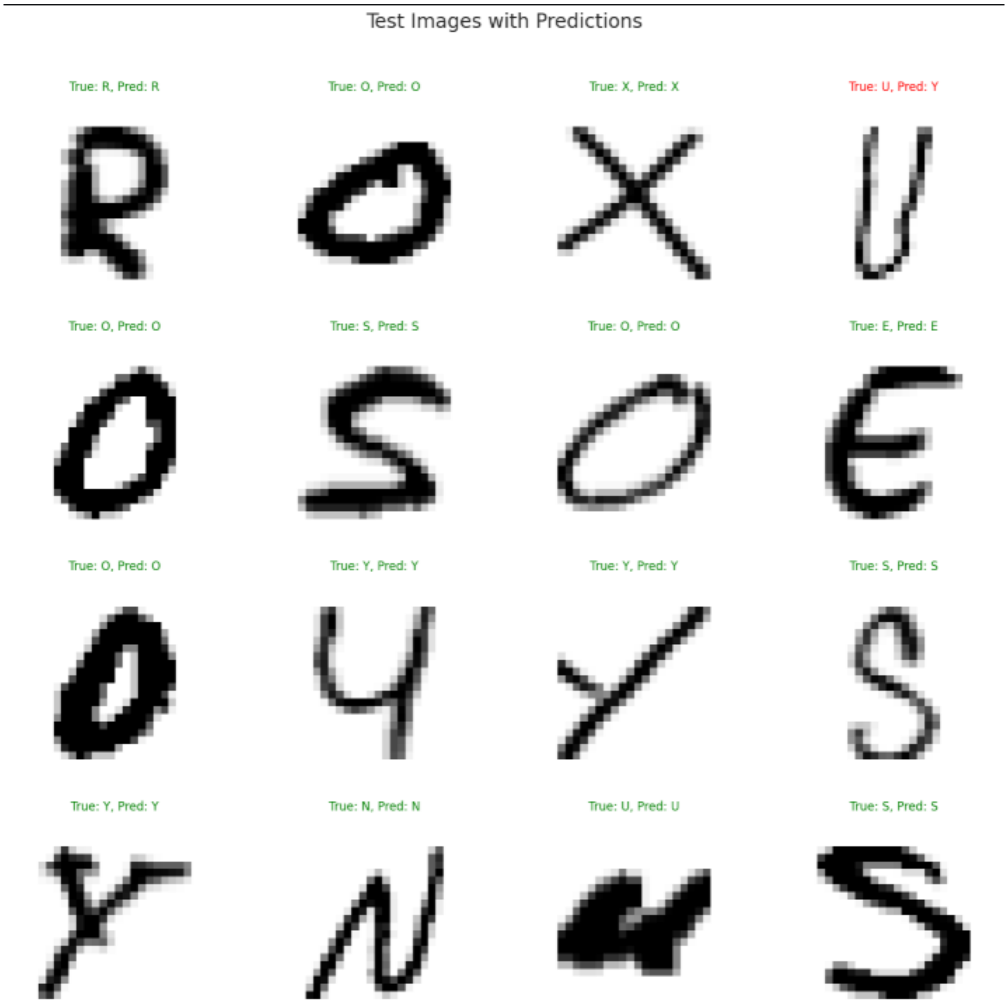
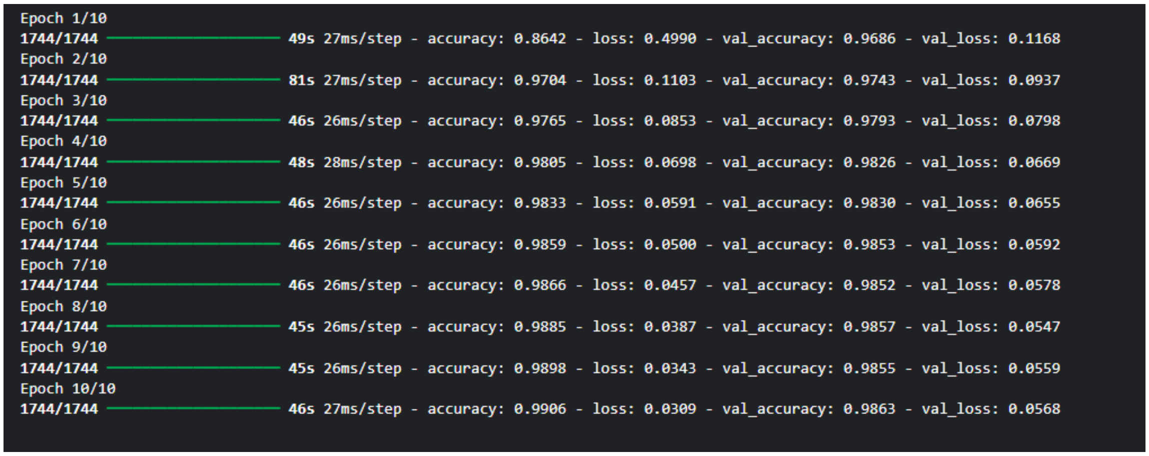
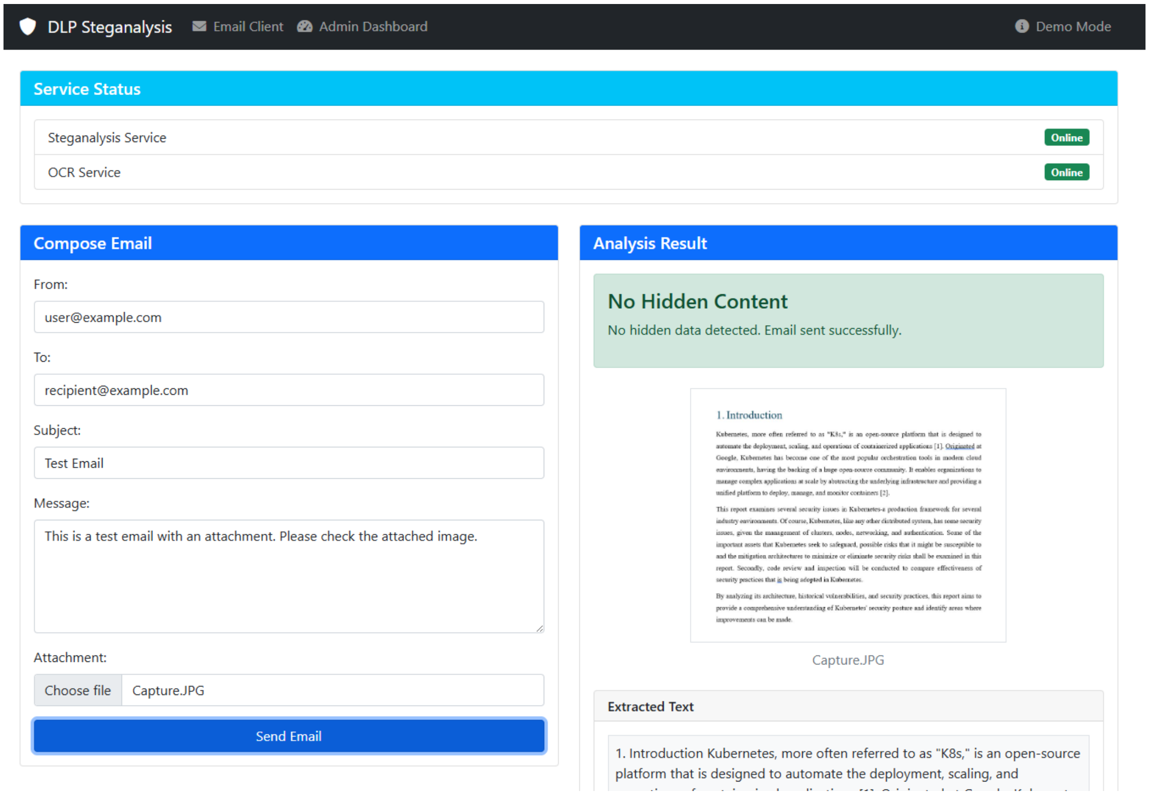


Figure 12: Test images with predections



```
Epoch 1/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 49s 27ms/step - accuracy: 0.8642 - loss: 0.4990 - val_accuracy: 0.9686 - val_loss: 0.1168
Epoch 2/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 81s 27ms/step - accuracy: 0.9704 - loss: 0.1103 - val_accuracy: 0.9743 - val_loss: 0.0937
Epoch 3/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 46s 26ms/step - accuracy: 0.9765 - loss: 0.0853 - val_accuracy: 0.9793 - val_loss: 0.0798
Epoch 4/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 48s 28ms/step - accuracy: 0.9805 - loss: 0.0698 - val_accuracy: 0.9826 - val_loss: 0.0669
Epoch 5/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 46s 26ms/step - accuracy: 0.9833 - loss: 0.0591 - val_accuracy: 0.9830 - val_loss: 0.0655
Epoch 6/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 46s 26ms/step - accuracy: 0.9859 - loss: 0.0500 - val_accuracy: 0.9853 - val_loss: 0.0592
Epoch 7/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 46s 26ms/step - accuracy: 0.9866 - loss: 0.0457 - val_accuracy: 0.9852 - val_loss: 0.0578
Epoch 8/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 45s 26ms/step - accuracy: 0.9885 - loss: 0.0387 - val_accuracy: 0.9857 - val_loss: 0.0547
Epoch 9/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 45s 26ms/step - accuracy: 0.9898 - loss: 0.0343 - val_accuracy: 0.9855 - val_loss: 0.0559
Epoch 10/10
1744/1744 ━━━━━━━━━━━━━━━━━━━━ 46s 27ms/step - accuracy: 0.9906 - loss: 0.0309 - val_accuracy: 0.9863 - val_loss: 0.0568
```

Figure 13: OCR model epochs



Service Status

Steganalysis Service	Online
OCR Service	Online

Compose Email

From: user@example.com

To: recipient@example.com

Subject: Test Email

Message: This is a test email with an attachment. Please check the attached image.

Attachment: Choose file Capture.JPG

Analysis Result

No Hidden Content

No hidden data detected. Email sent successfully.

1. Introduction

Kubernetes, more often referred to as "K8s," is an open-source platform that is designed to automate the deployment, scaling, and operations of containerized applications [1]. Originated at Google, Kubernetes has become one of the most popular orchestration tools in modern cloud environments, having the backing of a huge open-source community. It enables organizations to manage complex applications at scale by abstracting the underlying infrastructure and providing a unified platform to deploy, manage, and monitor containers [2].

This report examines several security issues in Kubernetes-a production framework for several industry environments. Of course, Kubernetes, like any other distributed system, has some security issues, given the management of clusters, nodes, networking, and authentication. Some of the important assets that Kubernetes seek to safeguard, possible risks that it might be susceptible to and the mitigation architectures to minimize or eliminate security risks shall be examined in this report. Secondly, code review and inspection will be conducted to compare effectiveness of security practices that is being adopted in Kubernetes.

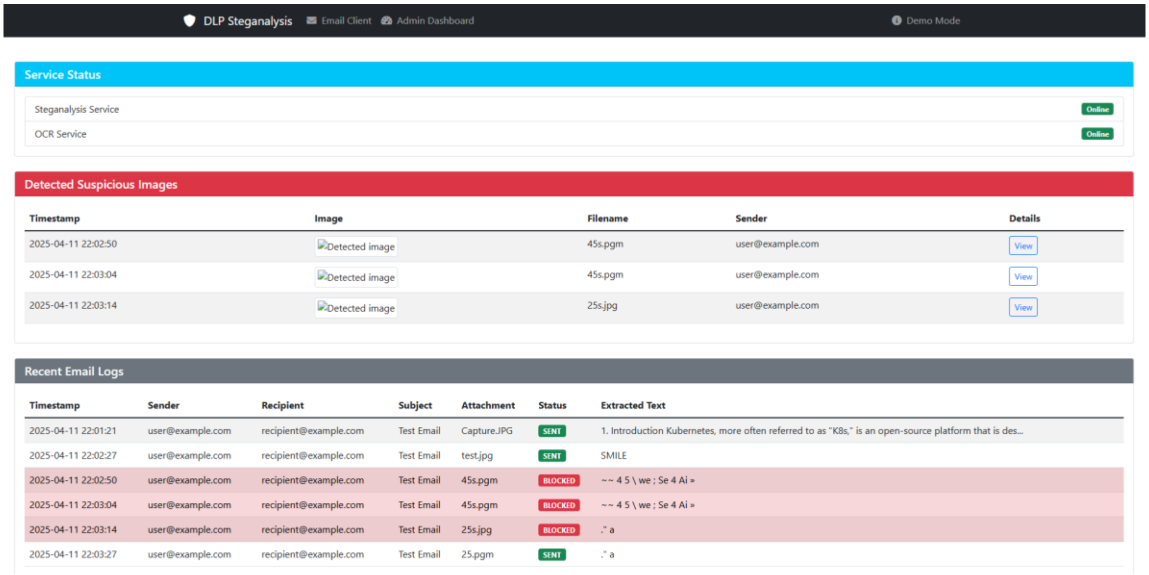
By analyzing its architecture, historical vulnerabilities, and security practices, this report aims to provide a comprehensive understanding of Kubernetes' security posture and identify areas where improvements can be made.

Capture.JPG

Extracted Text

1. Introduction Kubernetes, more often referred to as "K8s," is an open-source platform that is designed to automate the deployment, scaling, and

Figure 14:Email client dashboard



Service Status

Steganalysis Service	Online
OCR Service	Online

Detected Suspicious Images

Timestamp	Image	Filename	Sender	Details
2025-04-11 22:02:50	Detected image	45s.pgm	user@example.com	View
2025-04-11 22:03:04	Detected image	45s.pgm	user@example.com	View
2025-04-11 22:03:14	Detected image	25s.jpg	user@example.com	View

Recent Email Logs

Timestamp	Sender	Recipient	Subject	Attachment	Status	Extracted Text
2025-04-11 22:01:21	user@example.com	recipient@example.com	Test Email	Capture.JPG	SENT	1. Introduction Kubernetes, more often referred to as "K8s," is an open-source platform that is des...
2025-04-11 22:02:27	user@example.com	recipient@example.com	Test Email	test.jpg	SENT	SMILE
2025-04-11 22:02:50	user@example.com	recipient@example.com	Test Email	45s.pgm	BLOCKED	~~ 4 5 \ we ; Se 4 Ai »
2025-04-11 22:03:04	user@example.com	recipient@example.com	Test Email	45s.pgm	BLOCKED	~~ 4 5 \ we ; Se 4 Ai »
2025-04-11 22:03:14	user@example.com	recipient@example.com	Test Email	25s.jpg	BLOCKED	." a
2025-04-11 22:03:27	user@example.com	recipient@example.com	Test Email	25.pgm	SENT	." a

Figure 15: Admin dashboard