Design and Development of an Automated Steganography-Based Detection Tool for Hidden Malicious Content in Digital Images

[1]M. Deepika, [2]C. Sai Bhavani, [3]P. Varun Teja, [4]K. Koteswararao, [5]G. Abhignita

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Dept. of Computer Science and Engineering (IoT & CS incl BCT),
[1]Potti Sriramulu Chalavadi Mallikarjuna Rao College of Engineering & Technology, Vijayawada, A. P., India
[1]deepikamanepalli525@gmail.com, [2]saibhavanichandaluru@gmail.com,
[3]varunteja763@gmail.com, [4]koteswararaokandula41@gmail.com,
[5]abhignithagudela@gmail.com

***Abstract*— The increasing use of digital images across online platforms has led to a growing threat of steganography, where malicious data is concealed within image files to evade conventional security systems. This paper presents a Steganography Detection System designed to detect hidden data embedded in images downloaded from multiple sources, utilizing a dual-engine architecture that combines a Heuristic Engine based on classical detection methods and an AI Enhanced Engine leveraging advanced machine learning techniques, with all analysis performed within an isolated sandbox virtual environment to ensure safe and self-contained processing. The system employs seven detection algorithms — Least Significant Bit (LSB) Analysis, Discrete Cosine Transform (DCT) Analysis, Gradient Entropy Analysis, Metadata Analysis, File Signature Exploration, Hash Chain Verification, and HMAC Signature Verification — working in parallel to achieve comprehensive detection coverage, while detected stego images are quarantined and managed through user-controlled actions including force download or permanent deletion, with all scan events and outcomes recorded in a tamper-evident audit log for full traceability.**

***Index Terms*— Stego Image Detection,Sandbox Virtual Environment, Quarantine, Audit Log, Image Security, Digital Forensics, Autoscan.**

I. INTRODUCTION

These days, the use of images for purposes of communicating and transferring information is increasingly popular. Nevertheless, the image format can be exploited through a mechanism known as steganography, which enables the concealment of the transmitted message within the image, and the concealed data would be imperceptible by anyone examining the file visually. Hence, there arises a problem, since this kind of activity can lead to the exchange of confidential data or malicious content in a covert manner.Traditional computer protection systems such as anti-viruses and firewalls do not consider the presence of hidden information within an image file to be a threat. They are aimed at detecting more obvious and easily identifiable hazards, and as a result, this process of hiding the information within an image file can easily go unnoticed.

To combat this issue, this research proposes to create a program named StegoDetect. This intelligent system will continuously monitor certain folders for changes and conduct an analysis of the files contained within whenever necessary. There will be several methods used to detect the hidden data in an image file including Entropy and frequency analyses.From the above analyses, the software will determine whether an image is safe or not. If any steganographic data exists in an image, that image will be quarantined to eliminate risks while safe images will be placed in another file. The software also keeps safe log of all operations.The primary goal of this project was to develop an efficient mechanism for detecting hidden information in images and ensuring digital security. The use of automated monitoring and analysis together with user-friendly graphical user interface makes this project a practical solution.

II. LITERATURE REVIEW

[1] Oleksandr Kuznetsov Et al.,

This article presents a comprehensive study about the use of artificial intelligence particularly convolutional neural networks in image steganography detection. According to the authors, they evaluate the performance of the state of art of the SRNet model on multiple steganographic techniques like WOW, HILL, S-UNIWARD, and Spread Spectrum Image Steganography (SSIS). As per study results, SRNet works well on conventional techniques, but it fails significantly while detecting SSIS due to its noisiness in embedding. To overcome this limitation, the authors trained the SRNet model on SSIS-specific datasets, resulting in the better detection accuracy of SSIS. However, this enhancement has a downside, which slightly reduces performance in other techniques. The study makes it clear how training a model specifically to detect a certain method which is not the same as existing methods is highly useful. Further, the study makes it clear how using a universal detection model might not work for all steganographic methods.

[2] Awab Qasim Karamanji Et al.,

A comparative analysis of various deep learning models employed for hidden information detection in images is presented in this paper. The authors present CNN-based binary classifications for cover and stego images generated by algorithms like WOW, S-UNIWARD, and HUGO. One of the major contributions of the paper is a hybrid Deep-CNN framework that is able to detect different types of steganography. Data set used here has 10,000 images making it a good dataset for evaluation. The

outcomes reveal that the fine-tuned Deep-CNN model surpasses traditional methods despite a minor drop in accuracy while pursuing universality. The authors also provide various techniques which can be used for steganography while also mentioning how difficult it will be to create a universal detection system that works for different embedding methods.

[3] Zhiyi Wang Et al.,

This paper introduces a novel deep learning–based steganography method that leverages Transformer architecture for improved feature extraction and embedding performance. Unlike traditional CNN-based approaches, the Transformer model captures global dependencies in images, allowing for higher capacity and better-quality steganographic outputs. Additionally, the authors propose a recursive permutation–based image encryption technique to enhance the security of hidden data before embedding. The combined approach, referred to as TRPSteg, not only improves hiding capacity but also ensures stronger protection of the secret image. Experimental results demonstrate that the proposed method outperforms existing state-of-the-art techniques in terms of visual quality and security. The paper highlights the growing importance of combining deep learning and encryption strategies in modern steganography systems.

[4] Yaofei Wang (2022) Et al.,

This paper presents a novel steganographic framework called SparSamp, which leverages deep generative models such as large language models (LLMs), image diffusion models, and speech generators to achieve provably secure steganography (PSS). The key idea is to embed secret messages through message-driven sampling, where pseudo-random numbers derived from the message guide the sampling process without altering the original probability distribution. This ensures that the generated stego content remains statistically indistinguishable from normal AI-generated content. The paper also introduces a sparse sampling strategy to improve embedding rate and decoding accuracy while maintaining low computational complexity of O(1). Experimental results demonstrate that SparSamp achieves high embedding capacity, fast decoding, and strong security compared to traditional arithmetic coding-based methods. Overall, this work contributes an efficient, scalable, and secure approach to steganography in the era of AI-generated content.

[5] Yaofei Wang (2025) Et al.,

This paper extends the concept of steganography in AI-generated content by focusing on achieving high embedding capacity with low complexity using generative models like GPT-based LLMs and diffusion models. It proposes the SparSamp technique, which integrates message encoding into the sampling process while preserving the statistical distribution of the model outputs. The study highlights that traditional provably secure steganography methods suffer from either low embedding rates or high computational cost, whereas SparSamp balances both effectively. The method adaptively adjusts sampling intervals to ensure unique token selection, thereby enabling accurate message extraction. Experimental evaluation shows that the approach can embed large amounts of data (e.g., high bits per token in text and high payload in images) while maintaining realistic output quality. The paper concludes that SparSamp provides a practical and efficient solution for real-world steganographic applications across text, image, and audio domains.

[6] Othman A. Alrusaini

In this paper, we analyze the robustness of various deep learning models used for steganalysis, i.e., for detecting the presence of hidden data within transformed images. This analysis covers the performance of different models under distortions of various kinds (such as compression and scaling). We analyze the behavior of state-of-the-art models, such as EfficientNet, SRNet, ResNet, Xu-Net, and Yedroudj-Net, under transformations of input images and propose a range of novel evaluation metrics, including Perturbation Sensitivity, Degradation Rate, and Resilience Threshold, as additional tools for measuring robustness along with accuracy and F1-score. We find that the most efficient models, EfficientNet and SRNet, are characterized by hierarchical feature extraction and adaptive scaling, which makes them highly resistant to image distortions. In conclusion, we discuss limitations of our approach and suggest possible areas for further research.

[7] Mostafa A. Ahmad Et al.,

This article is a survey of steganalysis systems. Specifically, the steganalysis approaches are classified into traditional and deep learning–based approaches. Statistical analysis, signature-based detection and transform domain techniques (DCT, DWT, FFT) rely on designed features to detect hidden information, but struggling with advanced patterns usually seen on embedding. Recent approach which have been proposed to tackle the above limitations involves deep learning models, most importantly, Convolutional Neural Networks (CNNs) that automatically learn features which discriminate images. Models like SFRNet, GBRAS-Net and DRN have been discussed with improvements shown in detecting hidden data at different embedding rates. Low accuracy of the current circuits and non-existence of efficient training data remain problems. In light of this study, a dual-CNN framework (SA-CNN and MBC-CNN) has been proposed that aims at improving detection accuracy and classification of hidden content into malicious or benign categories.

[8] Yinlong Qian Et al.,

This paper surveys the limitations of traditional steganalysis approaches that depend on handcrafted feature extraction followed by classification using methods like SVM or ensemble classifiers. These approaches are highly dependent on feature design and fail to adapt to complex image distributions. To address this, the paper introduces deep learning as a new paradigm where feature extraction and classification are unified using Convolutional Neural Networks (CNNs). The proposed GNCNN model automatically learns hierarchical feature representations from raw images, capturing subtle stego noise that is difficult to detect using conventional techniques. The survey also highlights how deep learning models outperform traditional methods by leveraging large datasets and optimizing feature learning jointly with classification, leading to improved detection performance across modern steganographic algorithms.

[9] Yuanyuan Ma Et al.,

This paper presents a framework for deep learning-based image steganalysis that uses evolutionary algorithms in the training process. The authors focus on issues concerning the limited convergence, overfitting, and inefficient optimization of parameters in steganalysis networks. They propose a population-based optimization approach where several networks are instantly initialized with heterogeneous hyper-parameters through Xavier and Kaiming initialization. These network "individuals" are evolved in accordance with the detection accuracy from selection, crossover, and mutation operations. The framework, during training, detects stagnation stages in the network's development phase and strengthens the network by adjusting the parameters. Our experimental outcomes show that our proposed network outperforms Xu-Net and Yedroudj-Net both in terms of accuracy and convergence speed. Overall, this paper effectively demonstrates the power of combining evolutionary computation and deep learning for optimizing complex steganalysis.

[10] Anthony Rene Guzman

This thesis presents the implementation of deep convolutional neural network-based digital steganography for the purpose of hiding and recovering information in digital images. The project builds on a pre-existing CNN-based steganography model. Several architectural modifications and a custom gain(loss) function are proposed which includes image similarity metrics, namely SSIM, MSE and PSNR. The method improves the imperceptibility of the stego-images while concurrently ensuring the high recoverability of the hidden. In addition, it also discusses all earlier approaches of steganography including spatial domain and transform domain techniques and their comparison with neural network techniques. The experimental results demonstrate that the enhanced network outperforms the baseline model and generates visually indistinguishable images with improved quantitative performance. According to the results, deep learning has capable of development of steganography through automatic learning of features and optimization of trade-off between secrecy and quality.

Table 1 Literature Review Comparison Table

Author / Ref	Work Done	Algorithms Used	Advantages	Limitations	Accuracy
Oleksandr Kuznetsov Et al.	Evaluation of SRNet on multiple steganography techniques	CNN (SRNet), WOW, HILL, S-UNIWARD, SSIS	High performance on standard methods; improved SSIS detection with retraining	Poor generalization across all methods; dataset-specific training needed	High (varies per method; reduced for SSIS initially)
Awab Qasim Karamanji Et al.	Comparative analysis of deep learning models; hybrid Deep-CNN framework	CNN (Deep-CNN), WOW, S-UNIWARD, HUGO	Good performance on large dataset (10,000 images); better than traditional methods	Slight drop in accuracy for universal detection	High but slightly reduced for universality
Zhiyi Wang Et al.	Transformer-based steganography (TRPSteg) with encryption	Transformer, Recursive permutation encryption	Captures global features; higher capacity; better security	Higher computational cost; complex architecture	Higher than state-of-the-art methods
Yaofei Wang Et al. (2022)	SparSamp framework for provably secure Steganography using generative models	Generative models (LLMs, diffusion), Sparse sampling	High embedding capacity; O(1) complexity; strong security	Depends on generative model quality	High (not numerically specified)

Yaofei Wang Et al. (2025)	High-capacity steganography using SparSamp with improved sampling	Generative models (GPT, diffusion), adaptive sampling	Balances capacity & complexity; realistic outputs; accurate decoding	Still limited by model constraints	High embedding efficiency (not specified)
Othman A. Alrusaini	Robustness analysis of deep learning steganalysis models under distortions	EfficientNet, SRNet, ResNet, Xu-Net, Yedroudj-Net	Strong robustness (EfficientNet, SRNet); new evaluation metrics	Limited dataset & transformation s	EfficientNet & SRNet show highest performance
Mostafa A. Ahmad Et al.	Survey of steganalysis systems (traditional vs deep learning); proposed dual-CNN framework (SA-CNN, MBC-CNN)	Statistical methods, DCT, DWT, FFT, CNN (SA-CNN, MBC-CNN)	Improved detection & classification (malicious/benign) ; automatic feature learning	Low accuracy in existing systems; lack of training data	Improved over traditional methods (exact value not specified)
Yinlong Qian Et al.	Survey of traditional vs deep learning steganalysis; introduced GNCNN model	SVM, Ensemble classifiers, CNN (GNCNN)	Learns hierarchical features automatically; better detection than handcrafted methods	Requires large datasets; computationall y intensive	Higher than traditional methods (not specified)
Yuanyuan Ma Et al.	Deep learning steganalysis using evolutionary optimization	CNN, Evolutionary algorithms (selection, crossover, mutation), Xavier & Kaiming initialization	Better convergence; avoids overfitting; improved accuracy vs Xu-Net & Yedroudj-Net	Increased complexity; higher training time	Higher than Xu-Net & Yedroudj-Net
Anthony Rene Guzman	CNN-based steganography system for hiding & recovering images	CNN, SSIM, MSE, PSNR metrics	High imperceptibility; improved recovery quality; optimized secrecy-quality trade-off	Focuses more on steganography than detection	Improved over baseline model

III. SYSTEM ARCHITECTURE

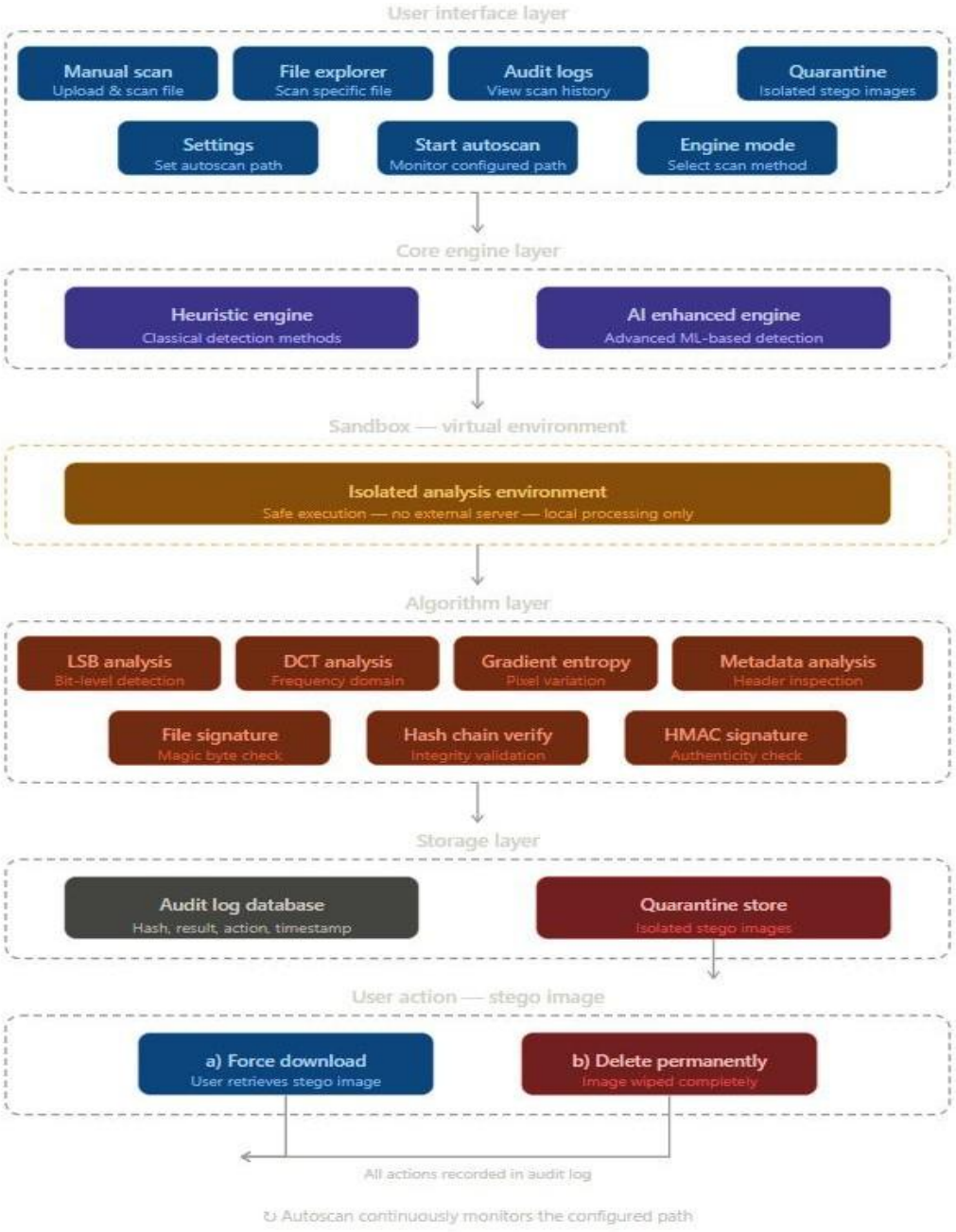

Fig 1 System Architecture

The system architecture represents the overall design and interaction between different modules involved in the steganalysis process. The architecture follows a layered approach to ensure modularity, scalability, and efficient data processing.

The proposed system consists of the following layers:

- Data Input Layer:

This layer is responsible for receiving input files from the user. The system supports multiple file formats including images, audio, video, and text. The uploaded files are validated and prepared for further analysis.

- Preprocessing Layer:

In this layer, the input files are processed to extract relevant data. For images, pixel values and color channels are extracted; for audio and video, signal and frame-level data are processed; and for text files, encoding patterns are analyzed. Preprocessing ensures that the data is in a suitable format for detection algorithms.

- Detection Layer:

This is the core layer of the system where multiple steganalysis techniques are applied. LSB analysis is performed to detect bit-level modifications, statistical methods analyze irregular patterns, and metadata inspection identifies anomalies in file properties. In addition, deep learning models are used to identify complex hidden patterns that are not easily detectable using traditional methods.

● Security (Sandbox) Layer:

To ensure safe processing, the system includes a sandbox environment where files are analyzed in isolation. This prevents potential threats such as malicious code execution and ensures system security during analysis.

● Result and Scoring Layer:

After analysis, the system generates a detection result along with a confidence score indicating the likelihood of hidden data. The results are presented in both visual and statistical formats to assist users in understanding the analysis.

IV. PROPOSED METHODOLOGY

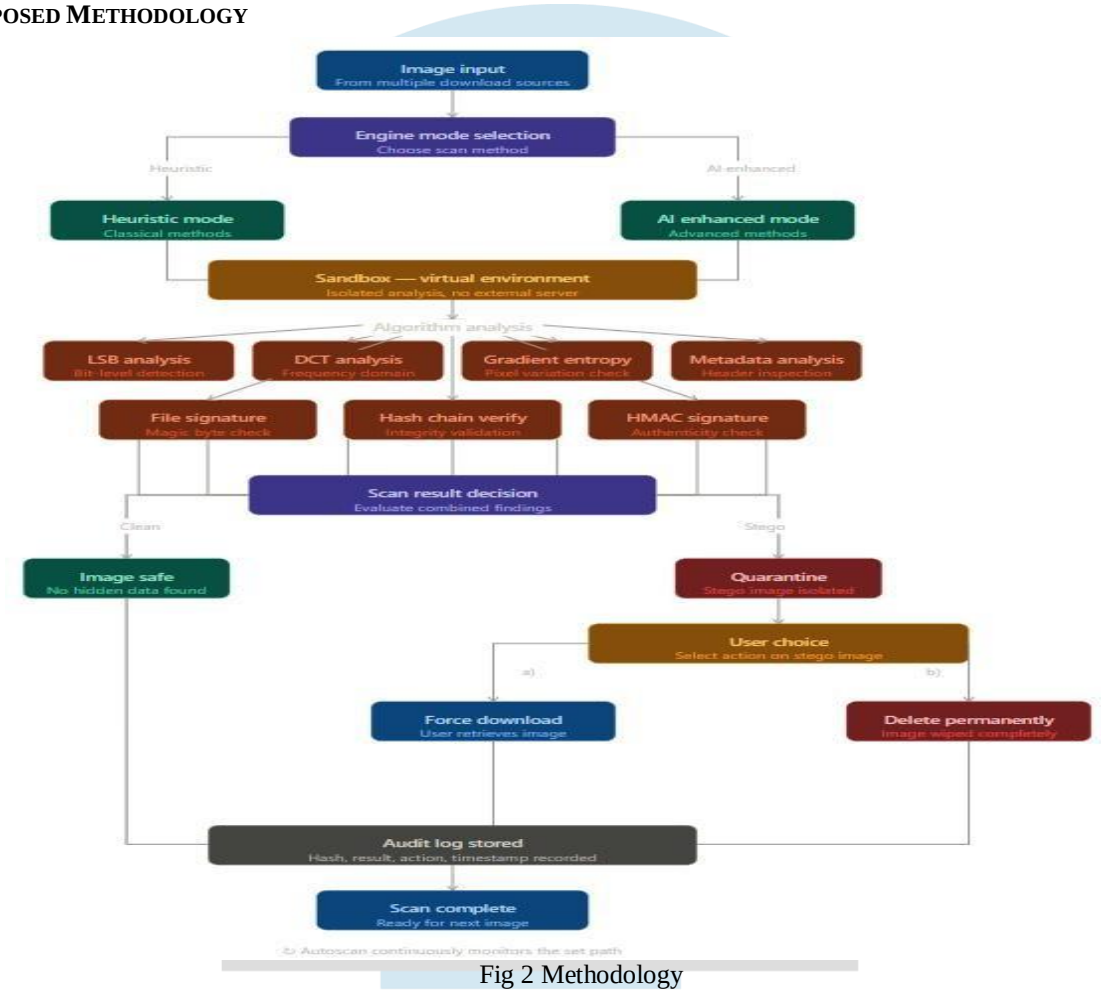

Fig 2 Methodology

Detection System of Steganography is meant to be an integrated system that will detect the presence of steganography in downloaded digital images. This detection system consists of five major parts that will be described in greater detail below: system architecture, the use of sandboxed virtual environment, use of dual engine detection system, algorithmic analysis of steganography, and management of detected steganography images.

4.1 *Sandbox Virtual Environment*

Image analysis in the proposed detection model occurs solely inside the isolated environment of a sandbox virtual machine. The sandbox represents a self-contained execution platform that entirely isolates the scanning process from any interaction with the underlying operating system and network infrastructure. As such, any potential maliciousness of the analyzed stego image will be incapable of executing any actions within the host system or establishing any unauthorized external communication channels while the image is being processed. In contrast to traditional server-based detection methods that send the image data to a remote endpoint for processing, the proposed system performs the analysis entirely inside the isolated sandbox. Thus, the security of the host environment and the analyzed images is preserved at all stages of the scanning process.

4.2 *Dual-Engine Detection Method*

Detecting steganography involves the dual engine approach where both classical methods of heuristic analysis and new methods of detection based on AI are employed.

4.2.1 *Heuristic Engine*

The Heuristic Engine applies classical methods of detecting distortions made in images via steganography. Such a type of detection is performed with the help of the rule-based system applying analysis of statistical and structural properties of an image to detect any alterations in it. This type of detection works rather effectively in cases when the aim is fast detection and is suitable for detecting such types of steganography as LSB steganography or changes in DCT coefficients. Heuristic detection operates deterministically analyzing an image with the help of certain algorithms to get its feature vector, which then is compared to a predetermined threshold value.

4.2.2 *AI Enhanced Engine*

The AI Enhanced Engine uses machine learning-based detection methods that rely on a vast number of clean and stego images to detect non-linear distortion patterns that are created by sophisticated steganography techniques that cannot be detected through traditional statistical analysis. This engine is specifically developed to counteract advanced steganography techniques like WOW, HILL, S-UNIWARD, and spread spectrum techniques that are explicitly aimed at reducing statistical artifacts and thereby avoiding detection by classical algorithms.

4.3 *Algorithm Level Analysis*

The seven steganography detecting algorithms run in parallel within the sandbox on each input image. The parallel running of the algorithms guarantees the thorough analysis of the image to ensure the absence of any false negatives due to limitations in any particular algorithm.

4.3.1 *Least Significant Bit Analysis (LSB)*

LSB analysis involves an analysis of the statistical distribution of the least significant bit plane of each image channel for anomalies associated with the LSB embedding steganographic method. The LSB distributions of natural or clean images are random, but those of LSB stego images are uniformly random as a result of replacing the natural bit planes with the embedded message bits. Various calculations involving the LSB histograms, chi-square values, and sample pair analysis are performed in the algorithm to measure LSB manipulation of the images.

A typical research paper on LSB steganalysis focuses on detecting hidden data by examining changes in the least significant bits of image pixels, assuming that steganography disturbs the natural bit distribution. It mainly relies on statistical irregularities in bit patterns to identify possible embedding.

$$P(LSB = 1) \sim P(LSB = 0) \qquad (1)$$

The drawback of this approach is that it only checks basic bit distribution and fails when advanced embedding techniques preserve statistical properties, leading to high false negatives.

$$D_{LSB} = \frac{1}{N}\sum_{i=1}^{N} |LSB_{OBSERVED} - LSB_{EXPECTED}| \qquad (2)$$

In this proposed paper, LSB deviation is calculated instead of simple probability, measuring the extent of variation from expected behavior. This approach is more effective because it quantifies subtle changes and, when combined with other features in a scoring model, significantly improves detection accuracy compared to traditional LSB-based methods.

4.3.2 *DCT Analysis*

DCT analysis is conducted in the frequency domain by decomposing the image into its frequency components and analyzing the statistics of DCT coefficients within image blocks. Steganographic manipulation in the DCT domain, which is utilized in JPEG steganography techniques, produces distortions in terms of deviation from the typical histogram distributions of DCT coefficients as well as dependencies between adjacent DCT coefficients. The algorithm determines deviations of DCT coefficients' histograms, calibration features, and transition matrices of Markov chains for DCT coefficients.

A typical research paper on DCT analysis focuses on detecting hidden data in compressed images by analyzing irregularities in frequency coefficients, as steganography often alters these values during embedding. It assumes that changes in DCT coefficients indicate possible data hiding.

$$\mathbf{F} = \sum |DT_{\text{OBSERVED}} - DT_{EXPECTED}| \qquad (3)$$

The drawback of this approach is that it is mainly effective only for JPEG or transform-domain images and may fail for spatial-domain steganography, limiting its applicability.

$$S_{total} = \omega_1 H + \omega_2 D_{LSB} + \omega_3 F + \omega_4 T + \omega_5 M \qquad (4)$$

In my system, DCT analysis is combined with other features like entropy, LSB deviation, and metadata checks in a unified scoring model, making it more robust and effective across multiple steganography types compared to standalone DCT-based methods.

4.3.3 *Gradient Entropy Analysis*

Gradient entropy analysis is aimed at analyzing the complexity and randomness of pixel intensity gradients within the image. The pixel gradients have certain distributions characteristic of natural image content, which becomes disrupted when steganographic embedding is performed. In the process, local gradients are determined using Sobel operators, and then gradient entropy maps are created for different parts of the image. Gradient entropy distributions are compared with normal entropy distributions in order to reveal potential evidence of steganographic activity.

A typical research paper on DCT analysis focuses on detecting hidden data in compressed images by analyzing irregularities in frequency coefficients, as steganography often alters these values during embedding. It assumes that changes in DCT coefficients indicate possible data hiding.

$$\mathrm{H} = \sum_{i=1}^{n} p_i log_2 p_i \qquad (5)$$

The drawback of this approach is that it is mainly effective only for JPEG or transform-domain images and may fail for spatial-domain steganography, limiting its applicability.

$$S_{total} = \omega_1 H + \omega_2 D_{LSB} + \omega_3 F + \omega_4 T + \omega_5 M \quad (6)$$

In my system, DCT analysis is combined with other features like entropy, LSB deviation, and metadata checks in a unified scoring model, making it more robust and effective across multiple steganography types compared to standalone DCT-based methods.

4.3.4 *Metadata Analysis*

The metadata analysis technique involves looking at the header data and EXIF tags present within an image file to detect anomalies which can be related to any manipulation and/or alteration through steganography. A genuine image file should include metadata fields which correlate with the format and information about the origin as well as creation time of the image. The algorithm checks the metadata fields for the presence of unusual field values, incomplete or missing mandatory fields, inconsistent metadata entries and unusual software signatures which may suggest the use of steganography tools on the image file.

A typical research paper on metadata analysis examines image header information (such as EXIF data) to detect inconsistencies or unusual tags that may indicate hidden content or manipulation. It assumes that steganographic tools often leave identifiable traces in metadata fields.

$$M = \begin{cases} 1 & if\ metadata\ is\ suspisious \\ 0 & otherwise \end{cases} \quad (7)$$

The drawback of this approach is that many steganography methods do not modify metadata, and attackers can easily remove or normalize metadata, making this method unreliable when used alone.

$$S_{total} = \omega_1 H + \omega_2 D_{LSB} + \omega_3 F + \omega_4 T + \omega_5 M \quad (8)$$

In my system, metadata analysis is used as an additional feature within a multi-factor scoring model, enhancing detection when combined with statistical and frequency-based methods. This improves overall effectiveness compared to traditional metadata-only approaches, which are limited in scope.

4.3.5 *File Signature Analysis*

File signature analysis is performed to confirm the correctness of the structure of each image file by analyzing the magic bytes, header, and internal structure of the file in accordance with the standards for the indicated file format. Some steganography programs may add minor inconsistencies to the structure of the file or add data blocks beyond the official file boundaries that cannot be detected by conventional image viewers. The program reads and analyzes the magic bytes of the file at the beginning of the file, analyzes the consistency of the internal structures of chunks for file formats like PNG and JPEG, and examines additional data located after the official end of the file boundary.

4.3.6 *Image Verification Using Hash Chains*

The hash chain verification process validates the integrity of an image using cryptography hash values by matching them with reference values, thus detecting any modification to image content. This verification process uses the algorithm to produce SHA-256 hash values both for the entire image file and its individual segments, forming a hash chain that will validate the integrity of the image at various granularities. When the calculated hash values differ from reference hash values and present some inconsistencies in the formed hash chain, then there is high probability that the image may have been altered using steganography.

- current_hash = sha(256(entry + previous_hash)

$$H_n = SHA256(E_n + H_{n-1}) \quad (9)$$

  - $\mathbf{H_n}$ = current log hash
  - $\mathbf{E_n}$ = current log entry
  - $\mathbf{H_{n-1}}$ = previous hash

4.3.7 *HMAC Signature Verification*

The HMAC Signature Verification step involves a keyed-hash message authentication code being used to authenticate and ensure that the integrity of each image file is intact. Unlike normal hash functions that only check for the integrity of the data against accidental or intentional alterations, the HMAC signature verification step ensures that the HMAC code generated is unique and can only be verified using the same secret key used to generate it. For each image, the algorithm computes an HMAC value that is then compared against a known HMAC value in order to ascertain whether any changes have occurred to the image.

4.4 *Scan Result Decision*

After the analysis of parallel algorithm scanning process, the output results of all seven detection algorithms are collected and analyzed by scan result decision module to yield a final binary decision result. The scan result decision module uses the process of weighted voting by considering the detection results and confidence level of the algorithms based on their detection power and reliability. If the total evidences exceed the threshold set by the system for detection, then the image is marked as a stego image, otherwise as clean image. Once classified as stego image, it will be sent to quarantine module for further management.

Table 2 Comparision Table for the proposed paper and for the research papers

Technique	Used in Previous Papers	Used in the proposed paper	Improvement
Entropy Analysis	Yes (basic detection)	Yes	Combined with other features
LSB Analysis	Yes (primary method)	Yes	Uses deviation instead of simple detection
DCT Analysis	Yes (JPEG only)	Yes	Integrated with other methods
Metadata Analysis	Rare	Yes	Extra detection layer
Trailing Data Check	Not common	Yes	Detects appended payload
Hybrid Model	Mostly single-method	Yes	Multi-feature fusion
AI-Based Detection	Used in few papers	Yes	Higher accuracy
Real-time Monitoring	No	Yes	Practical implementation
Secure Logging	No	Yes	Tamper-proof system

AI Based Detection:

The main characteristics of such a research paper would be the usage of machine learning models that will learn features of an image dataset and then will be able to recognize whether an image is stego or not.

A typical research paper on AI-based detection uses machine learning models trained on image features to classify images as stego or clean, aiming to improve detection accuracy over manual methods. It relies learned patterns from datasets rather that fixed rules.

$$P(\text{Stego}/\text{Features})$$

The drawback of this approach is that performance depends heavily on training data quality and may fail to generalize to new or unseen steganography techniques, leading to overfitting issues.

$$S_{total}=\omega_1 H+ \omega_2 D_{LSB}+ \omega_3 F+ \omega_4 T+ \omega_5 M + \omega_6 P(\text{Stego}) \quad (4)$$

In my system, AI-based detection is integrated with hueriistic features in a hybrid scoring model, combining learned patterns with statistical analysis. The improves robustness and accuracy compared to standalone AI methods by reducing overfitting and ensuring reliable detection across different types of steganography.

However, the issue of the research above is that its effectiveness greatly depends on the quality of training data sets used and might fail in cases where the steganography technique is not used before.AI-based detection will be implemented along with heuristic features in my system. It will help eliminate the problem described above and will ensure greater effectiveness and reliability of detection.

4.5 *Post-Detection Management*

4.5.1 *Quarantine*

Once the image is classified as stego image, the image will be automatically isolated to the quarantine store provided within the system, and stored there in order to prevent it from being accessed or used.

4.5.2 *User Action*

When the file is placed in quarantine, the user is presented with two possible actions. The “Force Download” button enables the user to download the stego image file to his local system for further inspection and legitimate uses. Meanwhile, pressing the “Delete Permanently” button triggers the system to completely erase the stego image file within the quarantine storage space and any other locations in the file system where the image was saved.

4.5.3 *Audit Log*

The audit log contains entries for all scans performed, the corresponding detection outcomes, quarantine procedures undertaken, and subsequent user actions. Each audit log entry includes details such as the image filename, its hash value, date and time of the scan, detection outcome, algorithms used in detecting the image, the engine mode, and the user's decision made.

4.6 *Autoscan*

Autoscan functionality allows the system to be used in an automatic scanning mode through the constant observation of a particular directory configured by the user that contains newly downloaded image files or newly created ones. Upon the recognition of a new image in the observed directory, the system automatically activates the scanning process without further user interaction. This functionality makes it possible to detect any potential steganography in images arriving in the environment that is under observation in real time.

V. RESULTS & DISCUSSIONS

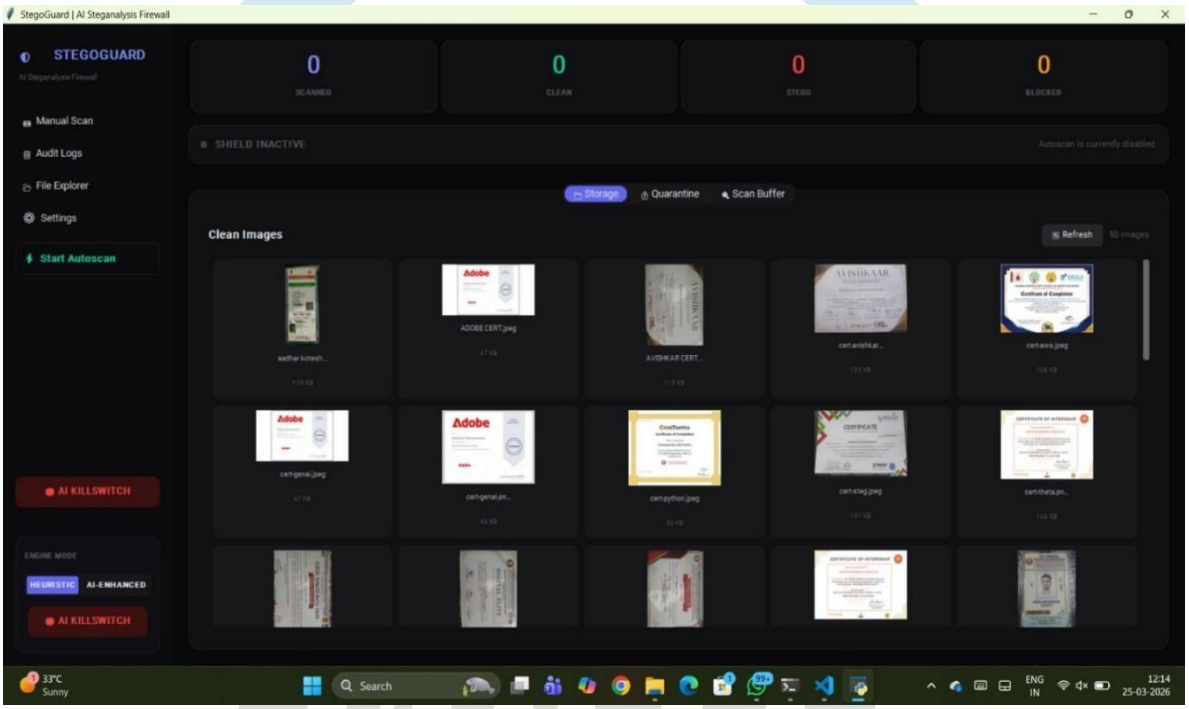

Fig 3 Initial State of StegoGuard Dashboard

This figure shows the initial interface of the StegoGuard steganalysis firewall before the scanning process begins. The dashboard displays system statistics such as total scanned images, clean files, detected steganographic threats, and blocked files, all of which are initially set to zero. The system indicates that the shield is inactive and autoscan is currently disabled.The interface includes multiple modules such as manual scan, audit logs, file explorer, and settings. A dedicated option to start autoscan is also provided, allowing users to initiate real-time monitoring. Additionally, the storage section displays previously verified clean images, ensuring easy access and management.

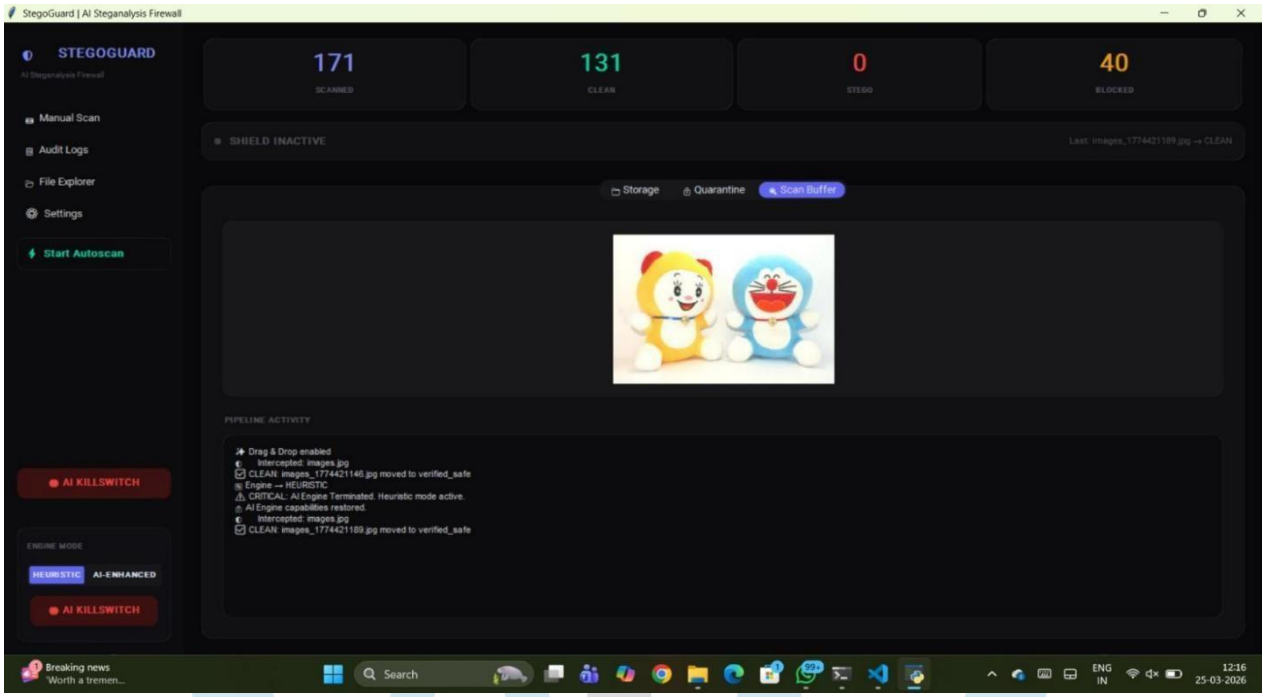

Fig 4 Real-Time Image Scanning and Classification

This is an image representing the real-time scanning feature of the StegoGuard system. From the dashboard, we can clearly see that the total scanned images are 171, out of which 131 are classified as clean and 40 are blocked based on the suspicious nature of the images. There are no steganography threats found in the scanned images.The scan buffer section shows the image being scanned, while the pipeline activity section records the entire process, including the interception, classification, and transfer to secure storage. The system is in heuristic mode, which is helpful in the efficient detection of anomalies without the need for using deep learning techniques.

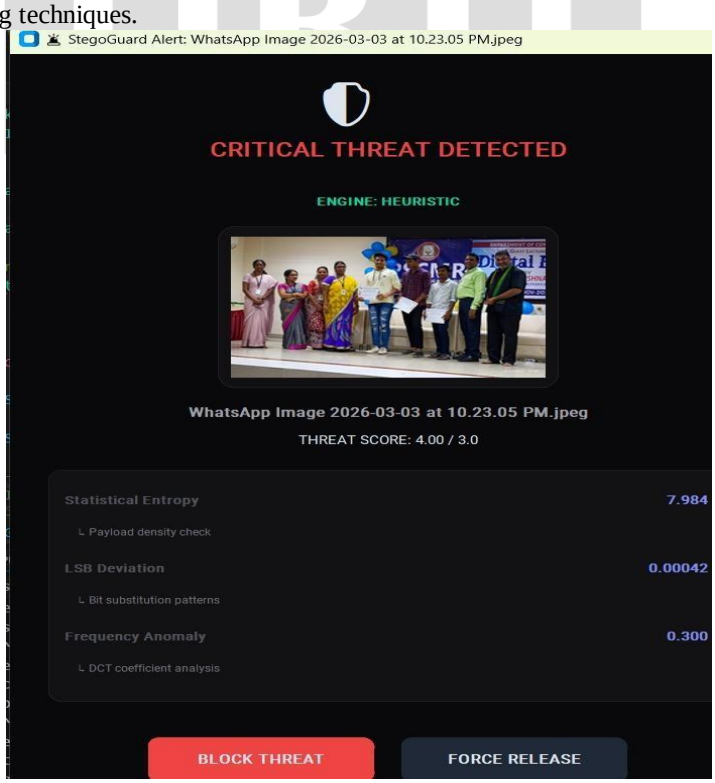

Fig 5 Detection of Steganographic Threat and Alert System

The below figure illustrates the identification of a implicit steganographic trouble by the system. After assaying the image, the system sends a critical alert, indicating the presence of suspicious content. The identification is carried out through heuristic analysis. The system offers detailed information regarding the identification criteria , including statistical entropy, LSB divagation, and frequence anomaly analysis. The trouble position is advanced than the threshold, indicating the presence of unusual patterns, which are associated with the identification of retired content. The system offers druggies a range of options, including the capability to block the trouble or release the train strongly. This illustrates the eventuality of the system to help image- grounded attacks.

VI. Acknowledgment

In our current project, we have implemented a software product called *StegoDetect*. Our software solution is intended to identify hidden data embedded in images to ensure increased protection of digital assets. Thus, “StegoDetector” can monitor chosen folders, automatically scan images, and recognize whether the analyzed image is secure or hides any information inside.

Such algorithms as entropy, least significant bits analysis, and frequency analysis are used for the detection of anomalies, which indicates the presence of steganography in the image. If there are suspicious changes identified in a picture, then the system moves it to a special quarantine folder. At the same time, if the image is found safe, it is kept in storage. All actions taken by our system are logged.

The benefit of the described project lies in the fact that it offers a convenient way to detect potential risks that can be missed due to the inability of regular security tools. This increases protection from hidden dangers significantly.

There are multiple ways to develop the presented project in the future. First of all, we should use more accurate AI-based models. Moreover, there can be implemented additional file formats like audio and video, as well as other security applications.

References

[1] O. Kuznetsov, E. Frontoni, K. Chernov, K. Kuznetsova, R. Shevchuk, and M. Karpinski, “Enhancing Steganography Detection with AI: Fine-Tuning a Deep Residual Network for Spread Spectrum Image Steganography,” *eCampus University*, 2024.

[2] A. Q. Karamanji, A. S. Ahmed, and A. F. Fadhil,“Comparative Deep Learning Models in Applications of Steganography Detection,” *Journal of Image and Graphics*, vol. 12, no. 3, 2024.

[3] Z. Wang, M. Zhou, B. Liu, and T. Li“Deep Image Steganography Using Transformer and Recursive Permutation,” *Entropy Journal*, 2024.

[4] Y. Wang, G. Pei, K. Chen, J. Ding, C. Pan, W. Pang, D. Hu, and W. Zhang,“SparSamp: Efficient Provably Secure Steganography Based on Sparse Sampling,” 2024.

[5] Y. Wang et al.,“SparSamp: Efficient Provably Secure Steganography Based on Sparse Sampling,” (Duplicate/Extended Version), 2024.

[6] Y. An and O. A. Alrusaini,“Deep Learning for Steganalysis: Evaluating Model Robustness Against Image Transformations,” *Frontiers in Artificial Intelligence*, 2025.

[7] M. A. Ahmad, E. Al-Qhtani, A. H. Samak, A. Ibrahim, M. Elloumi, and A. Ahmed,“Deep Learning-Based Steganalysis for Detection and Classification of Possible Hidden Content in Images,” *Fusion: Practice and Applications*, vol. 17, no. 2, pp. 377–393, 2025.

[8] Y. Qian, J. Dong, W. Wang, and T. Tan,“Deep Learning for Steganalysis via Convolutional Neural Networks,” *IEEE Transactions on Information Forensics and Security*, 2015.

[9] Y. Ma, X. Zhang, J. Wang, R. Jin, R. Nasimov, and H. Zhang,“Digital Image Steganalysis Network Strengthening Framework Based on Evolutionary Algorithm,” *Scientific Reports (Nature)*, 2023.

[10] A. R. Guzman,“Image Steganography Using Deep Learning Techniques,” M.S. Thesis, Purdue University, Fort Wayne, USA, 2022.