# Steganography Detection Toolkit

*A Digital Forensics Project Report*

**Submitted by:**

Group Members:
Taha Juzar 2022585
Ahmer Ayaz 2022070
Bashir Ahmed 2022646
Aurangzaib Bhatti 2022355

**Under the Guidance of:**

Dr. Shahab Haider
Department of Computer Science

**Ghulam Ishaq Khan Institute**
Department of Computer Science and Engineering
Topi, Pakistan

# 1 Introduction

Steganography is the art of concealing secret information inside regular, non-secret messages or files to prevent detection. It is different from cryptography because the presence of the concealed information is not revealed but instead hidden. In contemporary digital forensics, the detection of such concealed information is important in cybersecurity, criminal investigations, and data leakage prevention. This project targets the development of a Steganography Detection Toolkit, a forensic tool that can detect and recover concealed data in images, audio files, or documents by employing steganographic methods.

Figure 1: This is the original image which was tested under the steganography detection methodology

# 2 Objectives

The main objective of this project is to create and deploy a forensic software tool that can identify hidden information placed in digital files through steganography. The following are the specific objectives:

- Create a detection system that detects concealed payloads in popular file types like images, audio files, and documents.
- Apply analytical methods like entropy calculation, Least Significant Bit (LSB) pattern detection, and metadata anomaly detection to identify evidence of steganographic use.
- Support file carving functions by integrating with tools such as PhotoRec for the recovery of embedded or erased hidden content.
- Offer an intuitive graphical interface to enable investigators to upload files, conduct steganalysis, and examine or extract hidden data with low technical complexity.
- Provide compatibility and modularity to allow the toolkit to be easily expanded in the future to include further steganographic algorithms and file formats.

- Accompany digital forensic investigations by providing a consistent, automated means of analyzing possibly tampered or manipulated digital evidence.

# 3 Methods

The Steganography Detection Toolkit employs a multi-layered detection strategy that combines statistical analysis, file structure inspection, and payload extraction techniques. The primary methods used are as follows:

## 3.1 Entropy Analysis

Entropy measures the randomness within a file's data. Files that contain hidden or encrypted data often exhibit unusually high entropy. This toolkit calculates and visualizes entropy values to help forensic analysts identify anomalies that may suggest the presence of steganographic content.

## 3.2 Least Significant Bit (LSB) Pattern Detection

One of the most common steganographic techniques is embedding data into the Least Significant Bits of image or audio files. The toolkit scans for unusual LSB patterns, detects inconsistencies in pixel value distributions, and flags files with statistically improbable LSB noise.

## 3.3 Metadata Anomaly Detection

Many steganographic tools embed hints or even payloads in metadata sections. The toolkit inspects file headers, EXIF data (for images), and document properties for unusual or modified metadata that could indicate tampering or hidden content.

## 3.4 File Signature and Magic Number Verification

The toolkit uses `libmagic` to verify file signatures and detect format inconsistencies. Files whose content does not match their expected format (e.g., a JPEG file header on a non-image payload) are flagged for further inspection.

## 3.5 File Carving Integration with PhotoRec

To support deep analysis, the toolkit integrates with PhotoRec to perform file carving. This allows recovery of embedded or deleted content that traditional tools might miss, especially in cases where data has been hidden in unused sectors or appended to existing files.

# 4 Architecture of Toolkit

The Steganography Detection Toolkit is developed with a modular design that comprises four main components: the User Interface (UI), the File Handler Module, the Detection Engine, and the Extraction Layer. The User Interface offers a clean, easy-to-use graphical interface based on Python's Tkinter framework, which allows users to upload files, run analyses, and display results. It serves as the front-end for user interactions. The File Handler Module handles the uploaded files, with the help of the libmagic library to identify their proper format and verify they are compatible with the toolkit-supported file types (images, audio, and documents). At the center of the toolkit is the Detection Engine, which consists of different methods to detect covert data, including entropy analysis, LSB pattern detection, and inspection of metadata. Each method is written as an independent module to enable easy extension and customization. Lastly, the Extraction Layer is incorporated with PhotoRec for file carving, which makes it possible to recover concealed data that might not be apparent by using traditional analysis.

# 5 Implementation Details

The Steganography Detection Toolkit is written in Python, utilizing several libraries to aid its functionality. Tkinter is used to create the User Interface, offering a user-friendly interface where files can be uploaded, analysis started, and results viewed. For file handling, libmagic is used to check for valid file types and proper parsing, while OpenCV is used for image processing, particularly in extracting LSB patterns. The Detection Engine utilizes a variety of techniques in detecting anomalies in file randomness and LSB pattern detection to uncover steganography in image files. For the analysis of metadata, the toolkit examines file headers and properties in order to reveal evidence of tampering or hidden information. The toolkit also integrates PhotoRec for sophisticated file carving to support the recovery of partially deleted or concealed payloads. The modularity of the toolkit provides for each detection technique to be extended or replaced at a later time as new steganographic methods arise.

# 6 Results

The Steganography Detection Toolkit was tested with a number of test cases containing images with possible hidden information. Important parameters such as entropy, chi-square results, and visualizations of the LSB layer and RGB histogram were employed in order to measure the existence of hidden information.

## 6.1 Entropy Analysis

The entropy of a typical natural image is usually between 7.5–8.0, reflecting high randomness in pixel values. For images that were subjected to tests for possible hidden data, the entropy was much lower, at 5.2044. Lower entropy reflects non-random structure in the image, possibly because hidden data has a subtle effect on changing pixel values in a patterned way.

## 6.2 Chi-Square Test

A chi-square test was also conducted on the grayscale pixel distribution of the test image to determine how well it follows expected uniformity. The calculated chi-square statistic was 6,360,677.97 and the associated p-value was 0.0000. These findings reflect a strong deviation from expected randomness. A large chi-square value and a p-value approaching zero indicate the existence of organized modifications in the image, which is again a strong indicator that the image could include concealed data, for instance, by steganographic methods.

## 6.3 LSB Layer Visualization

Besides statistical tests, the visual inspection of the Least Significant Bit (LSB) layer offered additional proof of concealed data. Upon extraction and visualization of the LSB layer, patterns and inconsistencies could be seen, which signified potential embedding of concealed data. In steganographic-altered images, the LSB layer would show repetitive or uniform patterns, absent in natural, uncompromised images. This visualizing method accompanies the statistical procedures by providing a more concrete perception of the changes.

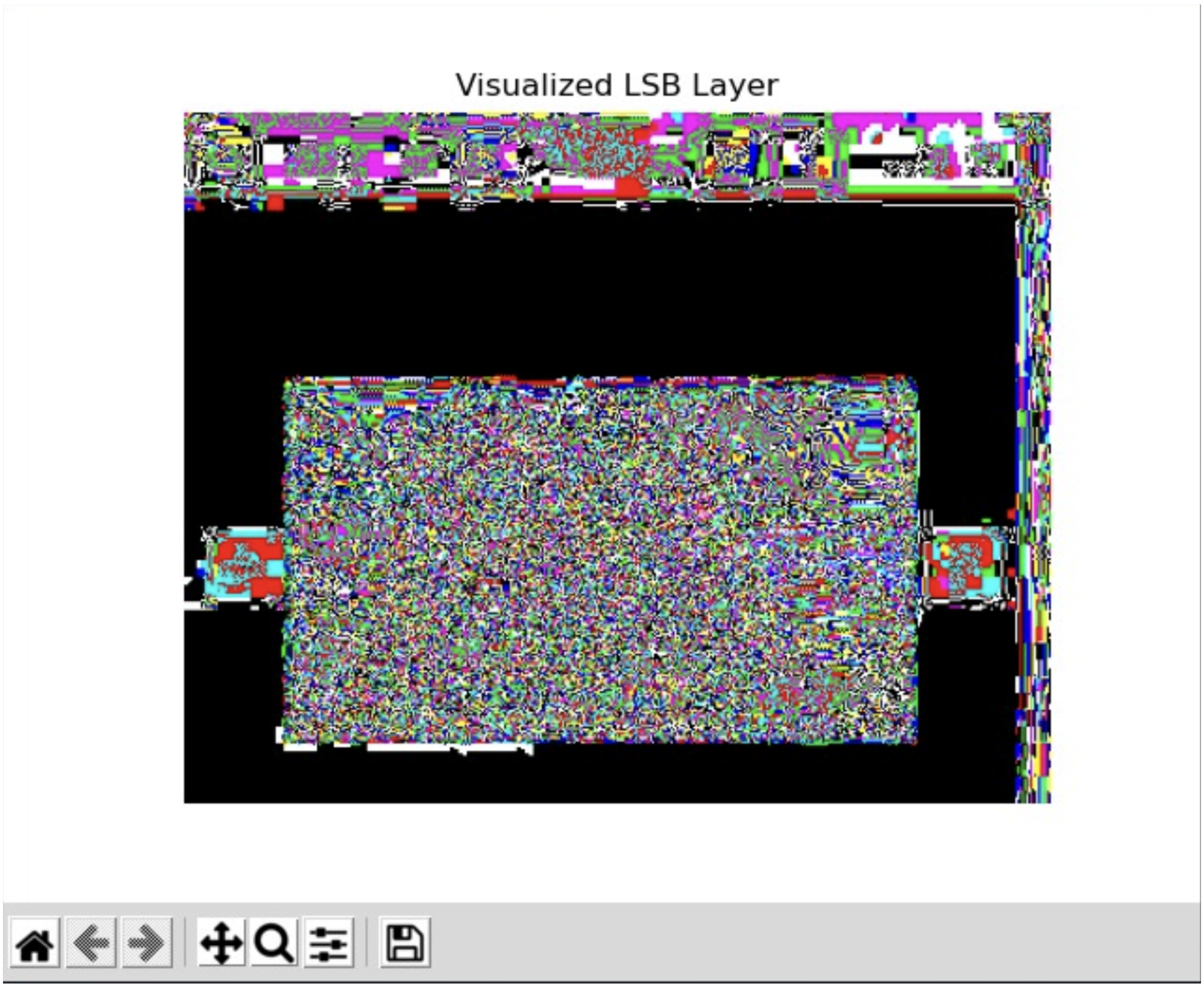

Figure 2: This image depicts the Least Significant Bit (LSB) plane of a digital picture. In normal images, the LSB layer is seen as approximately random noise because the LSBs make an insignificant contribution towards overall image color. With data embedded using LSB steganography, however, visible patterns, structured noise, or irregular distributions can occur.

### 6.4 RGB Histogram Analysis

Deeper analysis of the RGB histogram of the image also showed significant anomalies in the color channel distributions. In ordinary images that were not suspected of having hidden data, the RGB histograms would reveal an even distribution of pixel values over each color channel. However, on images suspected of having steganographic data, the histograms exhibited skewed distributions or unusual spikes in certain areas, indicating the presence of concealed information hidden in some color channels.

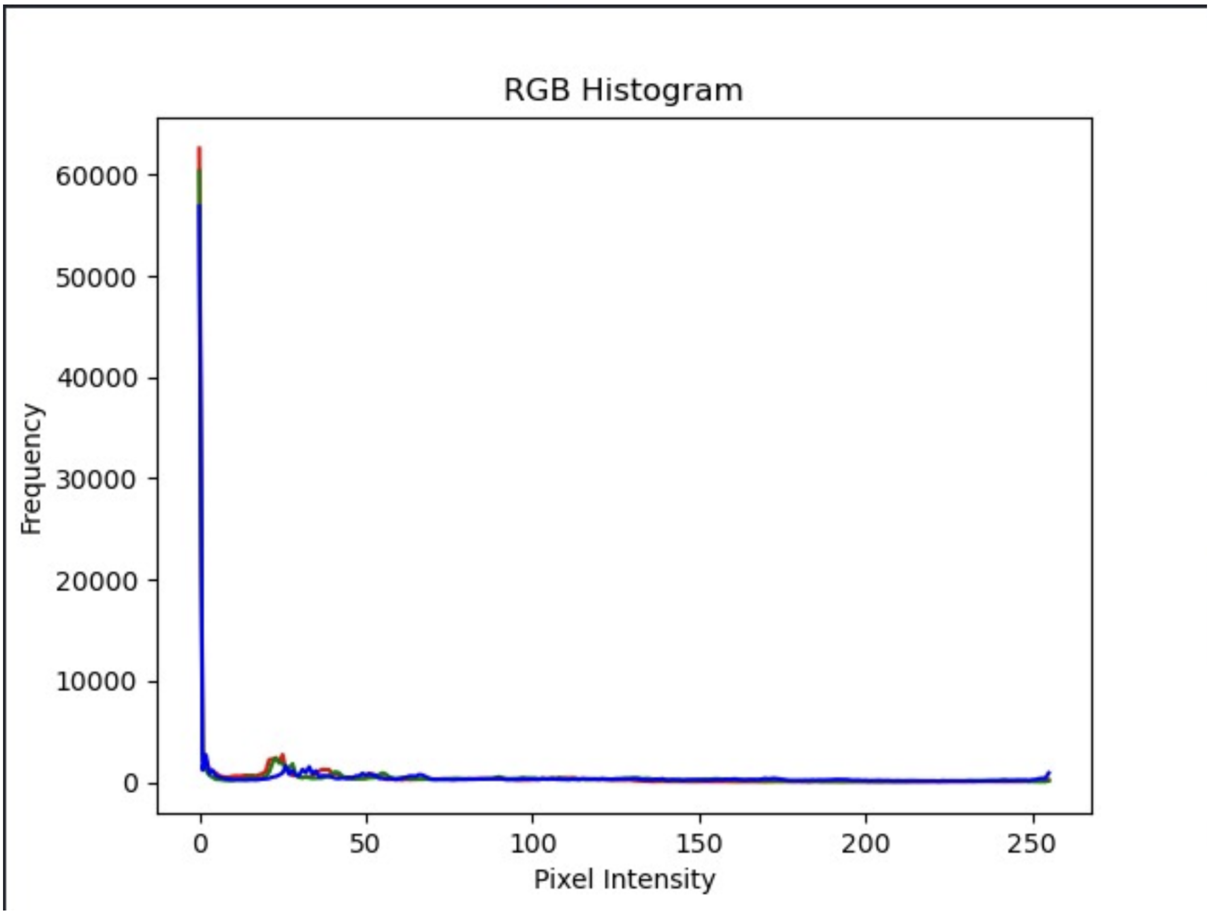

Figure 3:

## 7 Discussion

The findings of the Steganography Detection Toolkit are encouraging in terms of detecting hidden content in different file formats, especially images. By merging statistical tests such as entropy analysis and the chi-square test with visual techniques such as LSB layer extraction and RGB histogram analysis, the toolkit offers a multi-angled steganography detection approach. The statistical testing assisted in locating major anomalies within pixel randomness and distribution, and the visual approaches improved the toolkit's capacity for identifying concealed data more intuitively.

### 7.1 Challenges and Limitations

Despite the high performance of the toolkit, some challenges and issues arose when testing it. One of them is that the toolkit is sensitive to false positives. Although statistical techniques like entropy and chi-square tests work well, they sometimes report files that are not tampered with, particularly when the image itself contains natural noise or compression artifacts that change pixel values. To counteract this, additional refinement of the detection thresholds is necessary to lower false positives and enhance precision. Another constraint is file type dependency. The toolkit at present is mainly image-oriented, and although it also accommodates audio and document analysis, more work needs to be done to deal with more intricate file types such as video or compressed archives. As steganographic methods develop and new methods are discovered, ongoing updates and enhancements to the toolkit will be needed to keep it effective for different formats.

# 8 Conclusion

The Steganography Detection Toolkit effectively illustrates the real-world application of digital forensic methods to detect concealed data in multimedia files. Through the integration of statistical approaches like entropy and chi-square analysis with graphical tools such as LSB layer visualization and RGB histograms, the toolkit offers a multi-faceted approach to steganalysis. Its modular design and the fact that it integrates with external tools like PhotoRec make it a flexible and extensible solution for forensic investigators, capable of handling a wide variety of file types. The results obtained during testing further confirm the toolkit's ability to identify anomalies that indicate the presence of steganographic content, especially in image files. Although effective, the toolkit also identifies the necessity of continuous improvement—particularly in minimizing false positives and increasing support for more sophisticated file formats and embedding methods. Improvements in the future may involve the addition of machine learning for smart classification and the inclusion of real-time or network-based steganography detection capabilities. In general, this project provides a solid basis for a real-world forensic tool and is part of the continued efforts in obtaining digital evidence and detecting hidden communication channels.

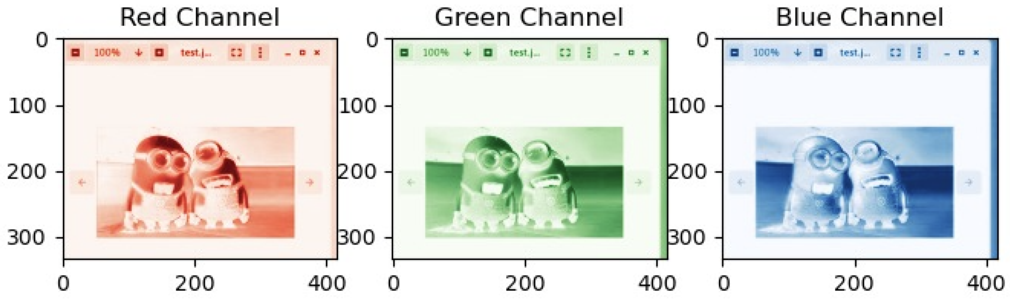

Figure 4: This image depicts the analysis of original picture after the implementation of Steganography detection toolkit