A MINI PROJECT REPORT ON

“Design and Develop a Tool for Digital Forensic of Images”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

(COMPUTER ENGINEERING)

UNDER THE GUIDENCE OF

Mr. Dattatray Modani

DEPARTMENT OF COMPUTER ENGINEERING
P.E.S MODERN COLLEGE OF ENGINEERING
PUNE 411005.

SAVITRIBAI PHULE PUNE UNIVERSITY
[2025 - 26]

Progressive Education Society's

Modern College of Engineering,

Shivaji Nagar, Pune- 411005.

Certificate

This is to certify that the following students of Computer Engineering of PES's. Modern College of Engineering have successfully completed their mini project in **Cyber Security And Digital Forensics** and designed the project entitled **"Design and Develop a Tool for Digital Forensic of Images"** under the guidance of the course instructor.

The Group Members are

Sr No.	Group Members	Roll Number
1)	Hemali Bharambe	41204
2)	Vishwajeet Londhe	41244
3)	Sahil Mate	41246
4)	Vaibhavi Mohite	41248

Internal Supervisor

Mr. Dattatray Modani

Head of Department

(Computer Engineering)

Prof. Dr. S. A. Itkar

Abstract

The rapid advancement of digital technology has made image manipulation and forgery increasingly common, posing serious challenges in verifying the authenticity of digital images. This project, titled "Design and Development of a Tool for Digital Forensic Analysis of Images," focuses on building a forensic application capable of analyzing digital images to detect tampering, extraction of metadata, and verification of authenticity. The proposed tool aids investigators and cybersecurity professionals in examining image evidence for integrity and reliability, which are critical aspects of digital forensics.

The system is designed using Python and its image processing libraries such as OpenCV and Pillow, along with metadata extraction modules like ExifRead or PyExifTool. It performs various forensic analyses, including metadata analysis, error level analysis (ELA), noise inconsistency detection, and copy-move forgery detection. These techniques help identify discrepancies caused by image manipulation, compression artifacts, or editing operations. The tool provides both visual and textual reports to assist investigators in understanding potential tampering regions and image authenticity indicators.

From a software testing and quality assurance perspective, the project involves thorough validation of each module to ensure accuracy and reliability of forensic results. Functional testing, performance testing, and image validation tests are conducted using diverse datasets of both genuine and manipulated images. Test cases are also created to evaluate the precision of forgery detection algorithms and ensure that false positives are minimized. The reliability of results is a key focus, making the tool suitable for use in legal and investigative contexts where evidence authenticity is critical.

In conclusion, this project demonstrates the integration of digital forensics, image processing, and software testing principles to develop a robust forensic tool for image authentication. By automating image examination and providing analytical insights, the system significantly aids forensic experts and investigators in identifying digital forgeries efficiently. This project not only highlights the importance of image forensics in cybersecurity but also showcases how systematic testing ensures the dependability and integrity of forensic tools in real-world applications.

Contents

Chapter 1

Introduction

1.1 Introduction

In the modern digital world, images play a vital role in communication, evidence presentation, and online information sharing. However, with the easy availability of advanced photo-editing tools, digital images can be altered or manipulated without leaving visible traces. This manipulation poses significant challenges in areas such as journalism, law enforcement, cybersecurity, and social media, where image authenticity is crucial. To address these challenges, the proposed project focuses on developing a Digital Image Forensic Tool that assists in detecting tampering and verifying the originality of digital images through various forensic techniques.

The tool is designed to analyze images using multiple forensic methods, such as metadata extraction, error level analysis (ELA), noise pattern detection, and copy-move forgery detection. By combining these methods, the system can identify whether an image has been modified, and if so, which areas show signs of alteration. The project also emphasizes the integration of software testing and validation processes to ensure that forensic results are accurate, reliable, and reproducible. This system aims to support digital investigators, cybercrime experts, and forensic analysts in establishing the credibility of image-based evidence.

1.2 Overview

This project provides a systematic approach to analyzing and verifying digital image integrity using automated forensic methods. The tool is developed using Python, leveraging libraries such as OpenCV, Pillow, and ExifRead for image processing and metadata extraction. These libraries enable the system to perform technical analysis, identify compression inconsistencies, and visualize potential tampering zones. The tool's user interface is designed to be intuitive, allowing users to upload images, run forensic tests, and view detailed analytical reports in both textual and graphical formats.

In addition to detection capabilities, the tool focuses on maintaining a modular architecture, separating the core image analysis algorithms, data handling, and reporting modules. This structure ensures flexibility and ease of maintenance. The project also follows a thorough software testing life cycle (STLC) to validate different functionalities of the system. Functional testing, boundary value testing, and performance testing are carried out to confirm that each forensic feature performs as expected. The outcome is a dependable and efficient image forensic application that can be utilized for both academic research and real-world

1.3 Problem Statement

With the rapid increase in digital media consumption, the authenticity of visual content has become a growing concern. Image manipulation techniques such as splicing, cloning, and retouching are often used to mislead viewers or falsify evidence. In legal investigations, fake images can distort justice, while in media, doctored images can spread misinformation rapidly. The absence of an easily accessible and automated forensic tool for image analysis creates difficulties for investigators and analysts who need to verify image authenticity quickly and accurately. Therefore, there is a strong need for a reliable digital forensic system capable of detecting manipulation and verifying the integrity of images.

The proposed project addresses this issue by designing and developing a **Digital Image Forensic Tool** that can perform multiple forensic analyses automatically. It eliminates the need for manual inspection, which is often time-consuming and prone to human error. The tool applies advanced forensic algorithms to analyze image structures, identify inconsistencies, and generate visual and textual reports. Additionally, the project ensures that the system undergoes proper software testing to maintain result accuracy and minimize false positives or negatives. The main goal is to provide a secure, efficient, and user-friendly solution for digital image authentication

1.4 Scope

The scope of this project covers both technical development and forensic investigation aspects, aiming to create a robust and intelligent tool capable of identifying digital image manipulation. The proposed Digital Image Forensic Tool is designed to help investigators, cybersecurity professionals, and researchers examine image files for authenticity, integrity, and possible tampering. With the growing number of digitally altered images being circulated online, the system provides an automated way to analyze, validate, and report findings efficiently. The tool focuses on delivering precise forensic insights that can be used as supportive evidence in cybercrime cases, media verification, and research studies involving digital imagery.

From a technical perspective, the project utilizes the Python programming language and popular image processing libraries such as OpenCV, NumPy, Pillow (PIL), and ExifRead. These libraries enable the tool to perform multiple forensic tasks, including Metadata Extraction, Error Level Analysis (ELA), Noise Inconsistency Detection, and Copy-Move Forgery Detection. Each of these functions serves a specific purpose in the forensic process. Metadata extraction reveals hidden details like the device used, date/time, and editing history of an image. ELA identifies variations in compression levels that often occur due to manipulation. Noise analysis detects abnormal pixel patterns, while copy-move detection helps identify duplicated regions within an image — a common technique in digital forgery.

Beyond the current implementation, the project's scope also includes future enhancement possibilities to improve its forensic capabilities and usability. In later versions, the tool could integrate machine learning and artificial intelligence algorithms for advanced forgery detection such as deepfake identification or semantic tampering analysis. Additionally, features like cloud-based data storage, secure evidence logging, and automated report generation in PDF format could be added to make the system more suitable for real-world forensic investigations. The tool can also be extended to handle video forensics or multiple image formats like RAW, TIFF, and WebP, increasing its practical applications.

Overall, the scope of this project is not limited to a single-purpose application but represents a complete digital forensic solution for image analysis. It demonstrates how software development, image processing, and forensic science can be integrated to address a growing cybersecurity need. By ensuring rigorous testing, robust design, and scope for future innovation, the project provides a valuable contribution to the field of Digital Forensics and Cybersecurity. It serves as both an educational model for understanding forensic algorithms and a practical tool for verifying digital evidence in real-world scenarios.

Chapter 2

Objectives

2.1 Objectives :

The main objective of this project is to develop a digital forensic tool capable of analyzing digital images to determine their authenticity, detect tampering, and extract hidden metadata. In today's digital world, image manipulation has become a common practice, which poses significant challenges in forensic investigations. This project aims to provide investigators and cybersecurity experts with an efficient solution for image analysis by integrating image processing techniques, forensic algorithms, and automated metadata extraction. The tool will be able to perform operations such as Error Level Analysis (ELA), Noise Detection, and Copy-Move Forgery Detection to identify signs of manipulation. Additionally, it will generate visual and textual reports that help users interpret the authenticity of an image easily.

Another key objective is to ensure software reliability and accuracy through systematic testing and quality assurance. The project includes designing and executing test cases to verify functional correctness, performance, and accuracy of detection results. The use of regular expressions will help validate file inputs and metadata formats. The ultimate goal is to create a user-friendly, accurate, and efficient forensic tool that can be further enhanced with AI-based detection in the future. Through this project, students will not only demonstrate technical proficiency in image forensics and software development but also showcase the importance of testing in ensuring the trustworthiness and reliability of digital forensic tools.

Chapter 3

System Specification

3.1 Library Used

This project makes extensive use of various Python libraries that provide built-in functionalities for image processing, data handling, visualization, metadata extraction, and forensic analysis. These libraries help reduce development time and ensure reliable performance while implementing complex image analysis algorithms. The following libraries have been utilized in this project:

1. OpenCV (cv2)

OpenCV (Open Source Computer Vision Library) is one of the most powerful and widely used libraries for computer vision and image processing tasks. In this project, OpenCV is used for reading, writing, and processing image files. It allows developers to perform operations such as resizing, color conversion (RGB to grayscale), edge detection, and region analysis. It also plays a key role in Error Level Analysis (ELA) and Noise Analysis, where pixel intensity variations are examined to identify tampered or altered areas in an image. OpenCV's speed and efficiency make it ideal for real-time forensic analysis.

2. Pillow (PIL)

Pillow is a modern fork of the original Python Imaging Library (PIL). It is primarily used for opening, manipulating, and saving images in different formats such as JPEG, PNG, BMP, and TIFF. In this project, Pillow helps perform pre-processing tasks like image resizing, cropping, and contrast adjustment. It also facilitates the implementation of ELA by saving image copies at a controlled compression level to detect hidden edits. Pillow provides easy integration with OpenCV and NumPy, which improves the overall efficiency of image analysis operations.

3. NumPy

NumPy (Numerical Python) is a core library for numerical computation and array manipulation. It represents images as multi-dimensional arrays (matrices), making it easier to perform mathematical operations such as filtering, averaging, and pixel intensity calculations. In this forensic tool, NumPy is used to process pixel-level data during forgery detection, calculate mean differences between image layers, and support operations in OpenCV and Pillow. NumPy also enhances performance by executing computations faster than native Python lists.

4. ExifRead / PyExifTool

ExifRead and PyExifTool are specialized libraries for extracting metadata (EXIF data) from digital images. Metadata provides important forensic information such as the camera model, image capture date, GPS location, editing software used, and other technical details embedded in image files. In forensic analysis, metadata can help identify inconsistencies between an image's properties and its claimed source. For example, if metadata shows an editing timestamp or software tag, it can indicate potential tampering. These libraries thus assist in establishing the authenticity and traceability of digital evidence.

5. Matplotlib

Matplotlib is a data visualization library used for displaying the results of image analysis graphically. In this project, it is used to plot the results of Error Level Analysis (ELA) and other detection algorithms. It helps in visually representing the areas of manipulation by displaying heatmaps or highlighting regions with unusual pixel intensity differences. This visual interpretation supports forensic experts in understanding the authenticity of images more intuitively and aids in report generation.

6. Tkinter / Flask

Tkinter (for desktop) or Flask (for web) is used to create the Graphical User Interface (GUI) or web-based frontend for the forensic tool. Tkinter allows users to browse and upload images for analysis, view metadata results, and display processed images with highlighted regions of tampering. Flask, on the other hand, can be used if the tool is implemented as a lightweight web application accessible via a browser. Both frameworks provide an easy way to interact with backend algorithms without requiring programming knowledge, making the system user-friendly and accessible to investigators.

7. Regular Expressions (re module)

The re (Regular Expression) module in Python is used for input validation and testing. It ensures that uploaded files are of correct image formats (e.g., .jpg, .png) and that extracted metadata follows valid patterns. Regular expressions are also used in testing to validate text-based outputs, filenames, and user entries. This enhances the overall reliability and robustness of the tool by preventing invalid data inputs during testing and execution.

8. Pandas (Optional)

Pandas is used for data management and analysis. Although optional, it can be implemented to store image metadata, test results, and analysis logs in structured tabular form. This helps maintain forensic records and aids in generating summary reports of multiple image examinations. It also supports exporting the data into formats like CSV or Excel for record-keeping and legal documentation.

3.2 System Requirements

The system requirements describe the minimum and recommended configurations needed to develop and run the digital forensic tool efficiently.

A. Hardware Requirements

- **Processor:** Intel Core i3 or higher
- **RAM:** Minimum 4 GB (8 GB recommended for faster processing)
- **Storage:** Minimum 500 MB of free space for installation and image datasets
- **Display:** 1024 × 768 resolution or higher
- **Graphics:** Integrated or dedicated graphics card supporting OpenCV rendering

B. Software Requirements

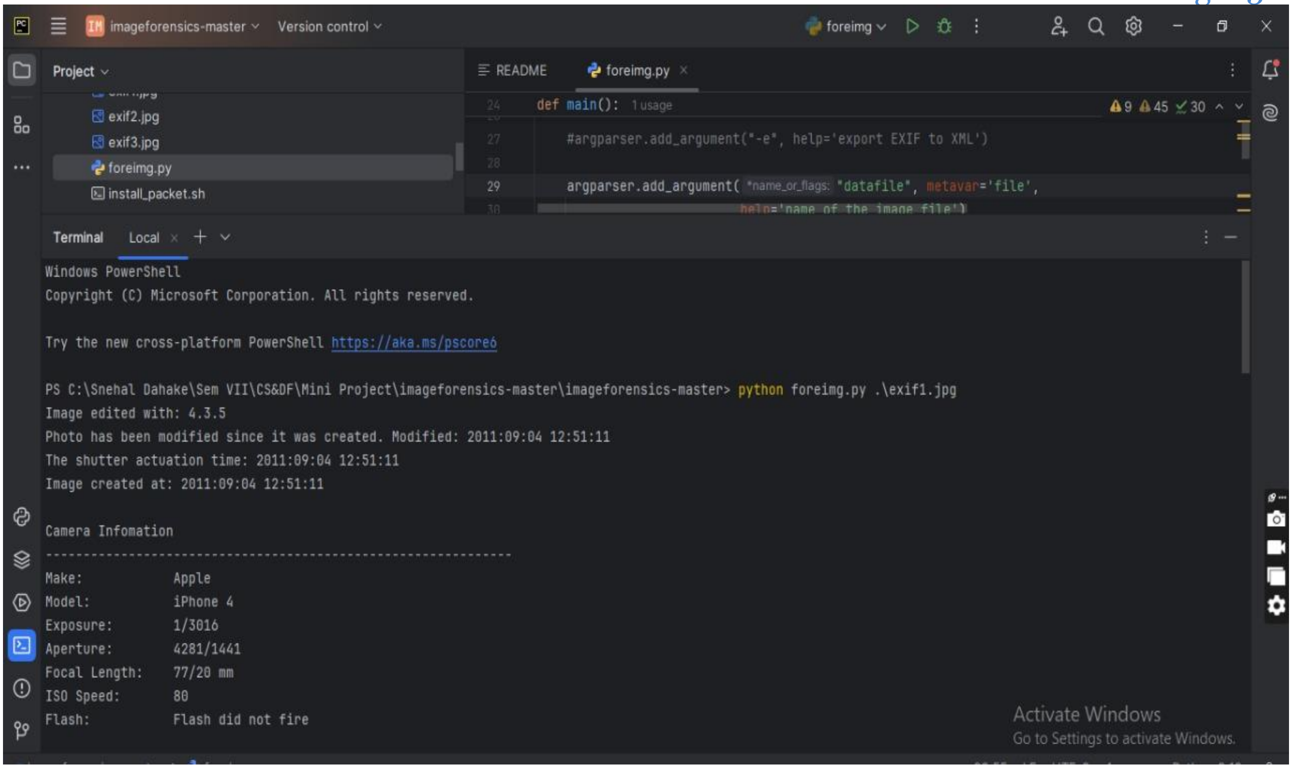
- **Operating System:** Windows 10 / 11, Linux Ubuntu, or macOS
- **Programming Language:** Python 3.8 or above
- **IDE / Editor:** PyCharm, Visual Studio Code, or Jupyter Notebook
- **Libraries:** OpenCV, Pillow, NumPy, Matplotlib, ExifRead, Regex, Tkinter/Flask
- **Database (optional):** SQLite or MySQL for storing image metadata or logs
- **Browser:** Google Chrome or Mozilla Firefox (if using a web-based interface)

C. Additional Requirements

- **Internet Connectivity:** Required only for installing Python libraries
- **Testing Tools:** PyTest or Unittest framework for quality assurance and validation
- **Dataset:** A collection of real and manipulated images for algorithm testing

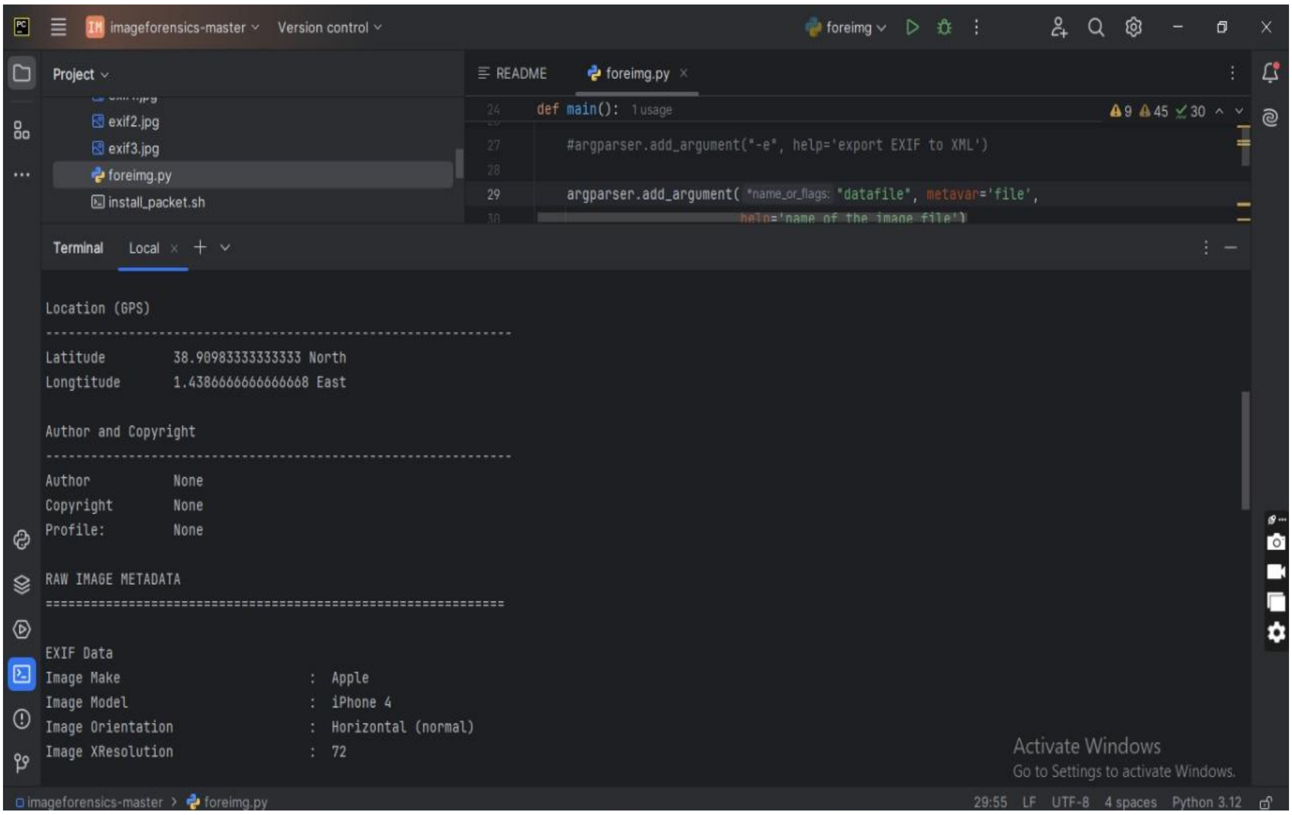
Chapter 4

OutPut Screenshots (GUI)



```python
def main():
    #argparser.add_argument("-e", help='export EXIF to XML')

    argparser.add_argument("datafile", metavar='file',
                           help='name of the image file')
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Snehal Dahake\Sem VII\CS&DF\Mini Project\imageforensics-master\imageforensics-master> python foreimg.py .\exif1.jpg
Image edited with: 4.3.5
Photo has been modified since it was created. Modified: 2011:09:04 12:51:11
The shutter actuation time: 2011:09:04 12:51:11
Image created at: 2011:09:04 12:51:11

Camera Infomation
--------------------------------------------------------------
Make:           Apple
Model:          iPhone 4
Exposure:       1/3016
Aperture:       4281/1441
Focal Length:   77/20 mm
ISO Speed:      80
Flash:          Flash did not fire
```



```python
def main():
    #argparser.add_argument("-e", help='export EXIF to XML')

    argparser.add_argument("datafile", metavar='file',
                           help='name of the image file')
```

```

Location (GPS)
--------------------------------------------------------------
Latitude        38.909833333333333 North
Longtitude      1.4386666666666668 East

Author and Copyright
--------------------------------------------------------------
Author          None
Copyright       None
Profile:        None

RAW IMAGE METADATA
=============================================================

EXIF Data
Image Make                          :  Apple
Image Model                         :  iPhone 4
Image Orientation                   :  Horizontal (normal)
Image XResolution                   :  72
```

Chapter 5

Conclusion

In conclusion, this project successfully demonstrates the design and development of a digital forensic tool capable of analyzing and verifying the authenticity of digital images. With the increasing rate of digital forgery and image manipulation in today's world, this tool serves as an important step toward ensuring the integrity of digital evidence. By using a combination of image processing, metadata extraction, and software testing techniques, the system effectively identifies tampering and provides reliable evidence to support forensic investigations. The integration of various Python libraries such as OpenCV, Pillow, NumPy, and ExifRead has enabled efficient and accurate image analysis, making the tool both practical and educationally valuable.

The project achieves its main objectives by detecting digital image manipulation through techniques like Error Level Analysis (ELA) and Noise Detection, while also retrieving EXIF metadata to uncover hidden image details such as camera model, location, and timestamp. The system provides both visual and textual reports, helping users interpret analysis results easily. Furthermore, the project emphasizes the role of Software Testing and Quality Assurance by implementing validation, functional, and performance testing to ensure consistent and reliable output. These testing measures not only enhance the tool's robustness but also make it suitable for academic learning and professional forensic applications.

From a technical perspective, the project highlights the importance of modular design and testing in developing secure and efficient forensic systems. The use of regular expressions, GUI frameworks, and open-source libraries demonstrates how modern technologies can be combined to create a flexible and scalable forensic solution. The tool's lightweight architecture allows easy integration with future enhancements, such as machine learning models for advanced forgery detection or cloud-based analysis for large-scale image investigations. Thus, the system provides a strong foundation for further research and development in the field of digital forensics and cybersecurity.

Overall, this project not only strengthens the understanding of image forensics and software testing concepts but also contributes to the growing field of digital evidence validation. It showcases how technology can be leveraged to protect digital authenticity and assist investigators in identifying manipulated or forged visual content. In the future, this project can be expanded with features such as AI-based forgery detection, report automation, and integration with forensic databases, further increasing its accuracy, usability, and real-world applicability in digital crime investigations.

Chapter 6

References

- Stallings, W. (2018). *Computer Security: Principles and Practice* (4th Edition). Pearson Education.
  – Provides foundational concepts on computer security and digital forensics.

- Nelson, B., Phillips, A., & Steuart, C. (2020). *Guide to Computer Forensics and Investigations* (6th Edition). Cengage Learning.
  – Explains key methods used in forensic analysis, including digital image forensics.

- Casey, E. (2019). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet* (4th Edition). Academic Press.
  – Discusses forensic methodologies for analyzing and preserving digital evidence.

- Farid, H. (2016). *Photo Forensics*. MIT Press.
  – Focuses specifically on techniques used to detect image manipulation and forgery.

- Mahdian, B., & Saic, S. (2009). "Using noise inconsistencies for blind image forensics." *Image and Vision Computing*, 27(10), 1497–1503.
  – Research paper detailing how noise patterns can help identify tampered regions in images.

- OpenCV Documentation. (2024). *Open Source Computer Vision Library*.
  Retrieved from https://docs.opencv.org/
  – Official documentation used for implementing image analysis and processing tasks.

- Python Software Foundation. (2024). *Python Standard Library Documentation*.
  Retrieved from https://docs.python.org/3/library/
  – Reference for Python libraries such as NumPy, Pillow, and Regular Expressions used in the project.

- ExifRead Documentation. (2024). *Metadata Extraction from Images in Python*.
  Retrieved from https://pypi.org/project/ExifRead/
  – Used to understand how to extract and interpret EXIF data from image files.

- Matplotlib Developers. (2024). *Matplotlib: Visualization with Python*.
  Retrieved from https://matplotlib.org/
  – Documentation for creating ELA visualizations and image analysis plots.

- Kaur, S., & Kaur, P. (2021). "Digital Image Forgery Detection Techniques: A Review." *International Journal of Advanced Research in Computer Science*, 12(3), 20–26.
  – A comprehensive review of existing image forgery detection algorithms and approaches.