LP4 (CSDF) Mini Project

Guide: Prof. Kimaya Urane

Name: Manish Godbole (31226)

Kaustubh Joshi (31233)

Aditya Kadu (31234)

Title: Digital Forensics of Image

Problem Statement:

Design and Develop a tool for Digital Forensics of Image.

Learning Objectives:

1. Understand the principles of digital forensics and its application in image analysis.
2. Learn to extract metadata (EXIF data) from images to gather relevant information.
3. Gain proficiency in calculating image hashes to ensure data integrity and authenticity.
4. Develop skills in validating image formats and checking for image integrity using basic analysis techniques.

Learning Outcomes:

By the end of this project, participants will have:

1. Ability to extract and analyse metadata from various image formats.
2. Proficiency in calculating and interpreting image hashes for integrity verification.
3. Skills in validating image formats and identifying potential file corruption.
4. Improved capability in performing basic image integrity checks through visual inspection and error analysis.

Theory:

1. Digital Forensics

- Digital forensics is a branch of forensic science that involves the recovery, preservation, and analysis of digital evidence.

- It aims to uncover and interpret data from electronic devices, including computers and mobile devices, in a manner that is legally admissible.
- In the context of images, digital forensics focuses on verifying the authenticity, integrity, and provenance of image files, which is crucial in various fields such as law enforcement, cybersecurity, and digital media.

2. Metadata (EXIF Data)

- Metadata refers to the supplementary information embedded within an image file that describes various attributes, such as camera settings, date and time of capture, and GPS coordinates.
- Exchangeable Image File Format (EXIF) is a standard format for storing metadata in image files, primarily JPEG and TIFF formats.
- Analysing EXIF data can provide vital clues about the image's origin, authenticity, and any modifications that may have occurred, making it an essential component of digital forensics.

3. Image Hashing

- Image hashing involves generating a unique fixed-size string (hash) from the content of an image using cryptographic algorithms such as SHA-256.
- This hash serves as a digital fingerprint for the image, allowing forensic analysts to verify its integrity over time.
- If the image data is altered, the resulting hash will change, indicating potential tampering.
- Hashing is a critical method for ensuring that evidence remains unaltered and can be trusted in forensic investigations.

4. File Format Validation

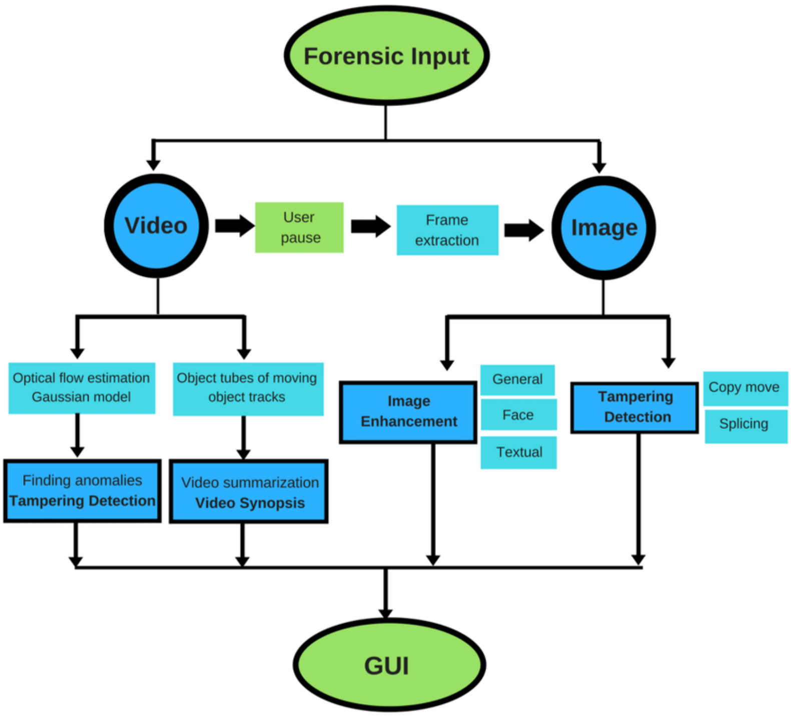
- Validating the file format is essential in digital forensics to ensure that the image file adheres to the expected standards and is not corrupted.
- This process involves checking the file's signature and structure to confirm its integrity.
- Tools and libraries like imghdr in Python can assist in identifying the actual format of an image file.
- Understanding file formats is crucial for forensic experts to differentiate between legitimate and potentially malicious or manipulated files.

5. Image Integrity Checks

- Checking the integrity of an image involves assessing its visual quality and ensuring that it has not been tampered with.
- Basic error level analysis (ELA) techniques can be employed to detect discrepancies in pixel values that may indicate manipulation.

- Additionally, displaying the image to verify its viewability can help forensic analysts identify any signs of corruption or modification.
- Ensuring image integrity is paramount in maintaining the authenticity of digital evidence.

System Architecture:


Forensic Input
Video
User pause
Frame extraction
Image
Optical flow estimation Gaussian model
Object tubes of moving object tracks
Image Enhancement
General
Face
Textual
Tampering Detection
Copy move
Splicing
Finding anomalies Tampering Detection
Video summarization Video Synopsis
GUI

The system architecture for the Digital Forensics of an Image using the imghdr library involves several components working together to provide forensics report. Here's an overview of the system architecture:

1. **Image Acquisition**

The first step in the digital forensics process involves gathering image files for analysis from various sources such as digital cameras, smartphones, or computer storage.

2. **Metadata Extraction**

Next, metadata extraction is performed to obtain valuable information embedded within the image files. This is achieved using the exifread library to read the EXIF metadata, which includes details like camera settings, date and time of capture, and GPS coordinates.

3. **Image Hash Calculation**

Following this, image hash calculation is conducted to ensure the integrity of the images. The SHA-256 hashing algorithm is implemented to generate a unique hash value from the image data, serving as a digital fingerprint that verifies the authenticity of the image.

4. File Format Validation

After hash calculation, file format validation is carried out to confirm that the images conform to expected standards and are not corrupted. The imghdr library is utilized to check the image format (e.g., JPEG, PNG) and ensure it is valid.

5. Image Integrity Check

Then, an image integrity check is performed to assess the visual integrity of the images. This involves opening the images with the PIL (Pillow) library for visual inspection, along with basic error level analysis to identify any signs of tampering or corruption.

6. Results Presentation

Finally, the results presentation step summarizes and compiles the findings from the forensic analysis. Extracted metadata, calculated hashes, validated formats, and integrity check results are compiled into a comprehensive report that can be used for further investigations or legal proceedings.

Dataset Description:

1. Image in png format

Methodology/Algorithm Details:

1. Image Acquisition

- **Objective**: Gather the image files that will be analysed for forensic purposes.
- **Method**: The images can be collected from various sources, such as digital cameras, smartphones, or computer storage. This step ensures that the images are ready for processing by the forensic tool.

2. Metadata Extraction

- **Objective**: Extract metadata from the acquired images to gather contextual information.
- **Process**: Utilize libraries like exifread in Python to read the EXIF metadata embedded within the image files. This metadata may include details such as camera settings, date and time of capture, and location data.

3. Image Hash Calculation

- **Objective**: Calculate a hash value for each image to ensure data integrity and verify authenticity.
- **Process**: Implement a hashing algorithm (e.g., SHA-256) to generate a unique hash for each image file. This hash will serve as a digital fingerprint, allowing analysts to detect any alterations to the image over time.

4. File Format Validation

- **Objective**: Validate the format of the images to ensure they conform to expected standards.
- **Process**: Use the imghdr library to check the image format (e.g., JPEG, PNG) and confirm that the files are not corrupted. This step helps in identifying potentially manipulated or unrecognized file types.

5. Image Integrity Check

- **Objective**: Assess the visual integrity of the images to detect any signs of tampering or corruption.

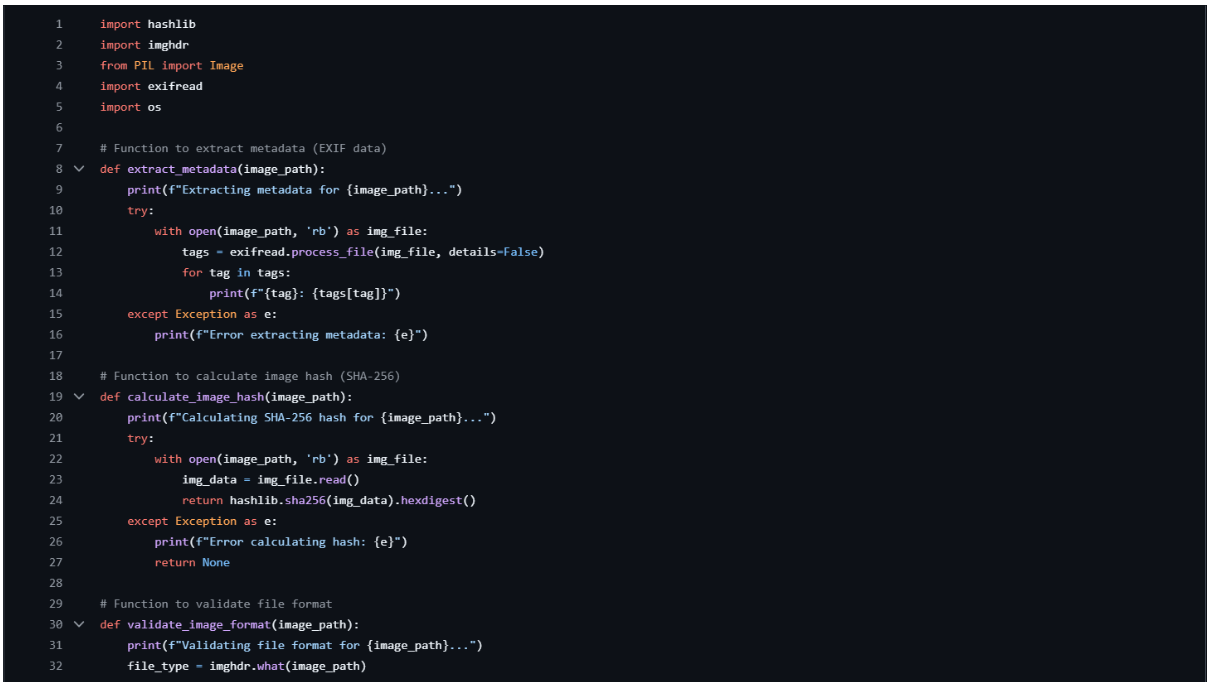
- **Process**: Open the image files using libraries like PIL (Pillow) to visually inspect them. Displaying the images allows analysts to confirm their viewability and check for basic corruption. Additionally, basic error level analysis can be applied to identify discrepancies in pixel values.

6. Results Presentation

- **Objective**: Summarize and present the findings of the forensic analysis.

- **Process**: Compile the extracted metadata, calculated hashes, validated formats, and integrity check results into a comprehensive report. This report can then be used for further investigations or legal proceedings.

Results:

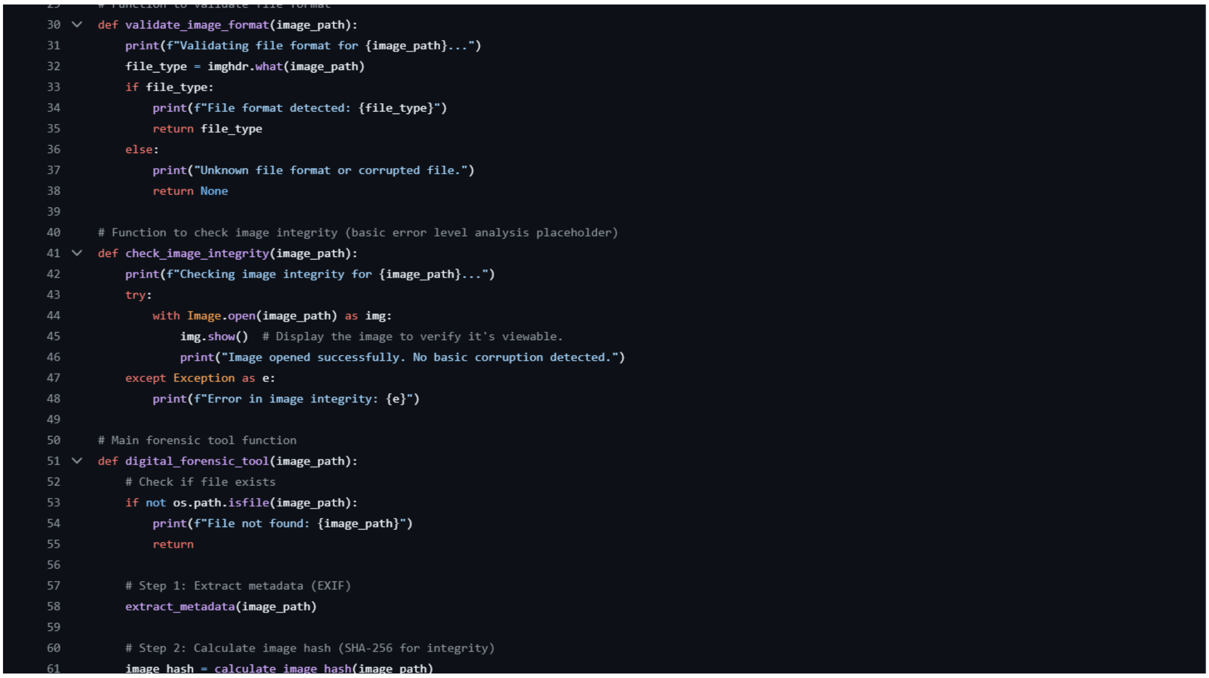


```python
import hashlib
import imghdr
from PIL import Image
import exifread
import os

# Function to extract metadata (EXIF data)
def extract_metadata(image_path):
    print(f"Extracting metadata for {image_path}...")
    try:
        with open(image_path, 'rb') as img_file:
            tags = exifread.process_file(img_file, details=False)
            for tag in tags:
                print(f"{tag}: {tags[tag]}")
    except Exception as e:
        print(f"Error extracting metadata: {e}")

# Function to calculate image hash (SHA-256)
def calculate_image_hash(image_path):
    print(f"Calculating SHA-256 hash for {image_path}...")
    try:
        with open(image_path, 'rb') as img_file:
            img_data = img_file.read()
            return hashlib.sha256(img_data).hexdigest()
    except Exception as e:
        print(f"Error calculating hash: {e}")
        return None

# Function to validate file format
def validate_image_format(image_path):
    print(f"Validating file format for {image_path}...")
    file_type = imghdr.what(image_path)
```



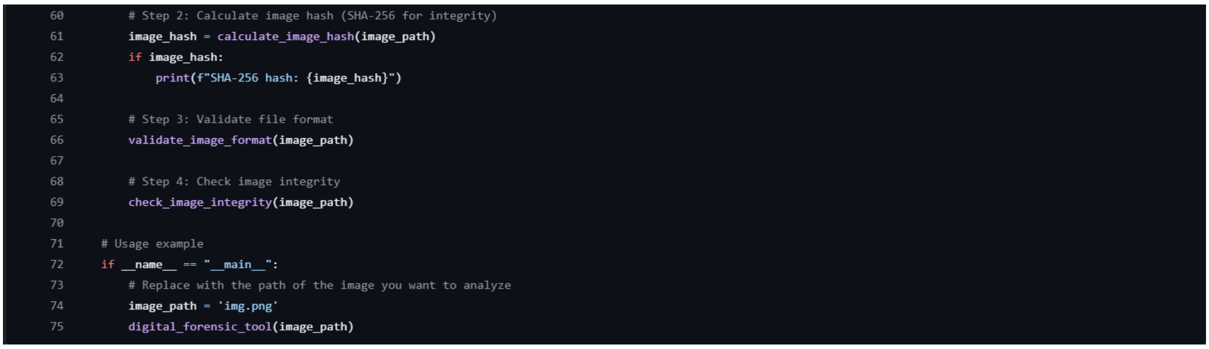
```python
# Function to validate file format
def validate_image_format(image_path):
    print(f"Validating file format for {image_path}...")
    file_type = imghdr.what(image_path)
    if file_type:
        print(f"File format detected: {file_type}")
        return file_type
    else:
        print("Unknown file format or corrupted file.")
        return None

# Function to check image integrity (basic error level analysis placeholder)
def check_image_integrity(image_path):
    print(f"Checking image integrity for {image_path}...")
    try:
        with Image.open(image_path) as img:
            img.show()  # Display the image to verify it's viewable.
            print("Image opened successfully. No basic corruption detected.")
    except Exception as e:
        print(f"Error in image integrity: {e}")

# Main forensic tool function
def digital_forensic_tool(image_path):
    # Check if file exists
    if not os.path.isfile(image_path):
        print(f"File not found: {image_path}")
        return

    # Step 1: Extract metadata (EXIF)
    extract_metadata(image_path)

    # Step 2: Calculate image hash (SHA-256 for integrity)
    image_hash = calculate_image_hash(image_path)
```



```python
    # Step 2: Calculate image hash (SHA-256 for integrity)
    image_hash = calculate_image_hash(image_path)
    if image_hash:
        print(f"SHA-256 hash: {image_hash}")

    # Step 3: Validate file format
    validate_image_format(image_path)

    # Step 4: Check image integrity
    check_image_integrity(image_path)

# Usage example
if __name__ == "__main__":
    # Replace with the path of the image you want to analyze
    image_path = 'img.png'
    digital_forensic_tool(image_path)
```

Analysis Conclusion:

The Digital Forensics of Images project successfully illustrates the application of various techniques and methodologies to analyze and verify the authenticity of image files. By implementing a structured approach that includes image acquisition, metadata extraction, hash calculation, format validation, and integrity checks, we have developed a comprehensive forensic tool capable of identifying potential tampering and ensuring data integrity.

The insights gained from metadata analysis provide valuable context regarding the images' origins and modifications, while the use of hashing algorithms confirms the integrity of the files. The validation of file formats and subsequent integrity checks further enhance the reliability of the forensic analysis.

This project underscores the importance of digital forensics in today’s increasingly digital world, where image authenticity is crucial in fields such as law enforcement, journalism, and digital media. Future work could involve enhancing the tool with more advanced analysis techniques, such as deep learning algorithms for detecting sophisticated image manipulations, thus expanding its applicability and effectiveness in real-world scenarios.