MODERN EDUCATION SOCIETY'S WADIA COLLEGE OF ENGINEERING,

PUNE, 411014.

DEPARTMENT OF COMPUTER ENGINEERING

.A PRELIMENERY REPORT ON

"Design and develop a tool for digital forensic of images"

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Computer Engineering

By

Sahil Naik S20111006
Sohan Chidrawar S20111013
Supreet Birajdar S20111017

Under the guidance of

Prof. S. K. Wagh

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-2023

CERTIFICATE

This is to certify that the project report entitles

"Design and develop a tool for digital forensic of images"

Submitted by

Sahil Naik	S20111006
Sohan Chidrawar	S20111013
Supreet Birajdar	S20111017

Are bonafide students of this institute and the work has been carried out by them under the supervision of Prof. S. K. Wagh and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering)

Prof. S. K. Wagh
Guide
Department of Computer Engineering

Dr. N. F. Shaikh
Head,
Department of Computer Engineering

Dr. Mrs. M. P. Dale
Principal,
Modern Education Society's College of Engineering Pune – 14

Place : Pune
Date : 29-10-2023

ACKNOWLEDGEMENT

The present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With thiswilling we joined this particular project. First of all, we would like to thank the supreme power the Almighty God who is obviously the one has always guided us to work on the right path of life.

We sincerely thank **Dr. N. F. Shaikh**, Head of the Department of Computer Science of Modern Education Society's college of engineering, for all the facilities provided to us in the pursuit of this project.

We are indebted to our project guide Prof. S. K. Wagh Department of ComputerScience of Modern Education Society's college of engineering college of engineering. We feel it's a pleasure to be indebted to our guide for his valuable support, advice and encouragement and we think him for his superb and constant guidance towards this project.

We are deeply grateful to all the staff members of CS department, for supporting us in all aspects.

We acknowledge our deep sense of gratitude to our loving parents for being a constant source of inspiration and motivation.

CONTENTS

Abstract

The manipulation of digital images has become very common in recent years. Thus, it is possible to cut, clone, and resize an image very quickly, which makes it challenging to validate the integrity and authenticity of images. Furthermore, digital images can be used by forensic experts in their forensic investigations. Digital image forensics (DIF) has emerged as an essential area of expertise focused on verifying the authenticity and integrity of digital files.

Examining the genuineness of images has emerged as a significant research domain. In our projects, we strive to extract the utmost possible information from images, employing readily available standard libraries designed for this specific purpose.

Our project focus on collection the information like Image Width, Image Length, Resolution Unit, Orientation, Date and Time, Exit offset and many more information.

Introduction

Currently, digital cameras have become very popular, and it is possible to find them in a variety of devices including smartphones, monitoring and surveillance cameras, etc. In addition, powerful image editing software has become very common. Due to such abundant availability of digital images, they can be used by forensic experts, for example, to help solve crimes. However, performing manual analysis of a large volume of digital images is a tedious task. In a typical research that lasts 6 to 18 months, forensic experts may evaluate over 100,000 digital files. In a forensic study, it was reported that in a database containing more than 300,000 image files and 1100 videos, only 148 images with illicit content (sexual abuse) were found.

Moreover, forensic decisions can be affected by several factors such as cognitive architecture and the brain, training and motivation, organizational factors, base rate expectations, irrelevant case information, reference material and case evidence. In this context, digital image forensics (DIF) is an area of knowledge focused on recovering and analyzing digital evidence in a criminal investigation process.

The primary objectives of this project are:

Data Collection and Acquisition: Develop mechanisms for collecting digital images from various sources, including cameras, social media platforms, and other digital media.

Image File Analysis: Examine the image file format, size, and compression. Suspicious compression artifacts can be indicators of manipulation.

Steganalysis: Detect information within images, considering the length, width, resolution, and orientation of the image, by analyzing the least significant bits or other steganographic methods.

Image Enhancement Detection: Identify traces of image enhancement or manipulation, considering how it affects the length, width, resolution, and orientation of the image.

Integration with External Tools: Integrate with external software or databases for cross-referencing information, including length, width, resolution, software, orientation, and date-time data.

Problem Statement

Design and develop a comprehensive digital image forensics tool aimed at enhancing investigative capabilities. This tool will empower users to analyze, authenticate, and uncover critical information within digital images, while ensuring the integrity, source, and any potential alterations are thoroughly examined. The tool will serve as an indispensable asset for forensic experts, law enforcement agencies, and researchers in their pursuit of accurate and reliable image analysis, aiding in the investigation and verification of digital image content.

Motivation

The motivation behind the proposed problem statement lies in the ever-increasing prevalence of digital images and the challenges associated with their authenticity. The rampant use of image manipulation software has led to a pressing need for tools that can discern between genuine and altered images. This is vital in legal and investigative contexts, where the integrity of image evidence is paramount. Furthermore, the tool's development is driven by the necessity to establish trust in an era of misinformation and deep fakes, protect digital identities, keep up with technological advancements, foster research and innovation, and facilitate interdisciplinary applications in various domains.

Objectives

The objective for the stated problem statement is to design, develop, and implement a robust digital image forensics tool with the following primary objectives:

1. **Authenticity Verification**: Develop algorithms and techniques to accurately determine the authenticity and origin of digital images, ensuring that manipulated or forged images are reliably detected.
2. **Information Extraction**: Create mechanisms to extract a comprehensive set of information from digital images, including metadata, dimensions, resolution, orientation, and other relevant attributes, to aid in investigative processes.
3. **User-Friendly Interface**: Design an intuitive and user-friendly interface to facilitate easy image analysis, enabling both forensic experts and non-experts to utilize the tool effectively.
4. **Real-World Utility**: Create a tool that meets the practical needs of investigators and legal professionals, enhancing their capabilities in authenticating digital image evidence for legal proceedings and investigations.

Theory

A Python script that utilizes the Pillow library (PIL) for image processing and metadata extraction. It performs several key functions to analyze and display image-related information. Below is a detailed analysis and explanation of the code for inclusion in a report:

1. **Library Installation and Import:**

The code begins with the installation of the Pillow library, which is commonly used for image processing in Python. It then imports essential modules, including Image from the Pillow library for image processing, TAGS from PIL.ExifTags for accessing EXIF metadata tags, and Image from IPython.display for image display within Jupyter Notebook.

2. **Image Display:**

The code defines the path to the image file 'test_image_2.jpg' and sets desired dimensions for the displayed image (width: 400 pixels and height: 300 pixels). It then uses the Image object from IPython.display to display the image in a Jupyter Notebook or IPython environment. This function allows users to visualize the image with the specified dimensions.

3**. Image Information Extraction:**

The code then proceeds to open the image 'test_image_2.jpg' using the Pillow library's Image.open() method. It extracts various basic metadata attributes, including the filename, image size, height, width, format, color mode, and whether the image is animated or contains multiple frames. These details are stored in a dictionary called info_dict.

4. **Printing Basic Image Information:**

A for loop iterates through the info_dict and prints the extracted basic image metadata, aligning the information in a readable format for user convenience.

5. **EXIF Metadata Extraction:**

The script further extracts Exchangeable Image File Format (EXIF) metadata from the image using the getexif() method. EXIF data often contains valuable information about the image, such as the date and time it was captured, camera settings, and more.

6. **Printing EXIF Metadata:**

The script iterates over all the EXIF data fields, translating the tag IDs to human-readable tag names using the TAGS dictionary. It then extracts the corresponding data for each tag. If the data is in bytes, it is decoded for better readability. The script prints out these EXIF metadata details in a well-organized format.

Libraries Used:

Python:

Python is the primary programming language. Python is a high-level, versatile, and widely used programming language known for its simplicity and readability. Python's ease of use and straightforward syntax, featuring indentation-based block structure, makes it an ideal choice for both novice and experienced programmers. It promotes rapid development and prototyping, enabling developers to translate concepts into code efficiently. Its versatile capabilities, combined with a focus on clean and maintainable code, have established Python as a prevalent language in various fields, reflecting its remarkable influence on modern software development.

Pillow:

Pillow, often referred to as the Python Imaging Library (PIL), is a powerful and popular Python library for working with images. It provides extensive support for opening, manipulating, and saving various image file formats.

Pillow, also known as the Python Imaging Library (PIL), is a comprehensive and widely used Python library tailored for image handling and manipulation. Its versatility extends to an extensive file format support, encompassing popular formats like JPEG, PNG, GIF, BMP, and TIFF, enabling seamless image opening and saving. Beyond basic operations such as resizing, cropping, rotating, and filtering, it facilitates advanced image manipulations and effects, making it an invaluable tool for image editing and enhancement. Pillow permits fundamental image functions like copying, pasting, and compositing, along with a suite of filters and enhancements for blurring, sharpening, color correction, and more. Users can overlay text, shapes, lines, and points on images, providing an easy way to add annotations or labels. Additionally, the library excels in converting images between different color spaces, extracting and handling EXIF metadata, extracting image dimensions, and working with various color modes. Pillow's ability to convert between image formats and its compatibility with other Python libraries like NumPy enhance its utility, while its user-friendly API and cross-platform compatibility make it accessible and efficient for a broad range of image-related tasks. Finally, Pillow enjoys an active development community, ensuring that it remains up-to-date and responsive to evolving Python versions and image formats.

Source Code:

Importing Libraries

In [15]:
```
!pip install Pillow
```

```
Requirement already satisfied: Pillow in c:\users\predator\anaconda3\lib\site-packages (9.4.0)
```

In [16]:
```
from PIL import Image
from PIL.ExifTags import TAGS
```

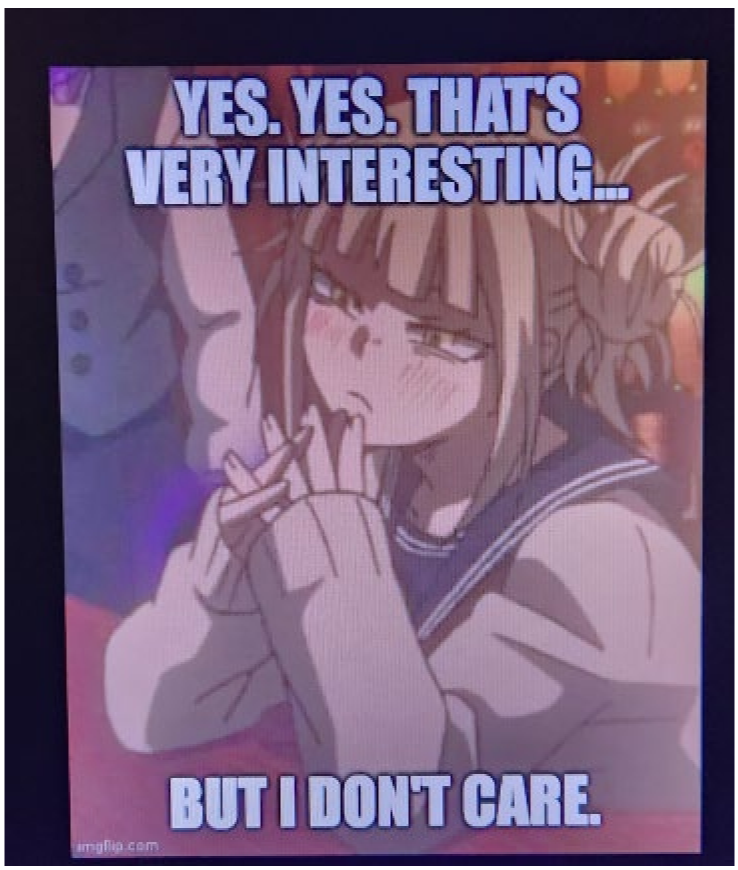
Displaying Test Image

In [25]:
```
from IPython.display import Image

# Define the path to the image file
image_path = 'test_image_2.jpg'

# Set the desired width and height for the displayed image
width = 400  # Replace with the desired width in pixels
height = 300  # Replace with the desired height in pixels

# Display the image with the specified width and height
Image(filename=image_path, width=width, height=height)
```



Extracting Metadata from Test Image

In [17]:
```
# path to the image or video
imagename = "test_image_2.jpg"

# read the image data using PILLOW
image = Image.open(imagename)
```

In [18]:
```
# extract other basic metadata
info_dict = {
    "Filename": image.filename,
    "Image Size": image.size,
    "Image Height": image.height,
    "Image Width": image.width,
    "Image Format": image.format,
    "Image Mode": image.mode,
    "Image is Animated": getattr(image, "is_animated", False),
    "Frames in Image": getattr(image, "n_frames", 1)
}

for label,value in info_dict.items():
    print(f"{label:25}: {value}")
```

```
Filename                 : test_image_2.jpg
Image Size               : (4000, 3000)
Image Height             : 3000
Image Width              : 4000
Image Format             : JPEG
Image Mode               : RGB
Image is Animated        : False
Frames in Image          : 1
```

In [19]:
```
# extract EXIF data
exifdata = image.getexif()
```

In [20]:
```
# iterating over all EXIF data fields
for tag_id in exifdata:
    # get the tag name, instead of human unreadable tag id
    tag = TAGS.get(tag_id, tag_id)
    data = exifdata.get(tag_id)
    # decode bytes
    if isinstance(data, bytes):
        data = data.decode()
    print(f"{tag:25}: {data}")
```

```
ImageWidth               : 4000
ImageLength              : 3000
ResolutionUnit           : 2
ExifOffset               : 226
Make                     : samsung
Model                    : SM-S901E
Software                 : S901EXXU6CWH3
Orientation              : 6
DateTime                 : 2023:10:27 16:42:11
YCbCrPositioning         : 1
XResolution              : 72.0
YResolution              : 72.0
```