APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 133-152
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Android JPEG File Carving with Geotag Analysis

Tam Jia Cherng[1], Nurul Azma Abdullah[1]*

[1] *Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: azma@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.008

Article Info

Received: 17 July 2024
Accepted: 16 October 2024
Available online: 15 December 2024

Keywords

Digital Forensics, Mobile Forensics, Android, JPEG, file carving, geotag.

Abstract

In today's digital age, recovering lost JPEG images from Android devices is crucial for preserving memories and supporting investigations. However, the lack of seamless integration between file carving and geotag analysis resulted in increased complexity of workflow and decreased productivity. This project developed an "Android JPEG File Carving with Geotag Analysis" tool to address this need. Using an object-oriented approach, the tool integrates file carving and geotag analysis techniques, implemented with Python. The target users for the tool are general consumers who want to recover their JPEG files from their Android smartphone's physical image and even individuals interested in digital forensics, particularly mobile forensics. The tool successfully recovers JPEG files, extracts geolocation metadata, verifies integrity through hash comparisons, and generates comprehensive reports. These features streamline digital forensic analysis, offering a robust solution for both general consumers and forensic enthusiasts. The tool's user-friendly design, effective file carving and geotag analysis functionality, and other features such as hash generation and visualized map highlight its significance in mobile forensics.

1. Introduction

In today's digital age, mobile devices have become essential tools for our daily communication, media consumption, and personal data storage. With the number of pictures or photos taken and stored on our smartphones increasing every day, the ability to recover the images in case of some accidents has become increasingly important as these images are our important memories, but they can also serve as evidence in some cases, such as an investigation. This project focuses on the development of an "Android JPEG File Carving with Geotag Analysis" tool designed to cater to the needs of a broad range of users, such as general consumers who want to recover their JPEG files from their Android smartphone's physical image and even individuals interested in digital forensics, particularly mobile forensics. Digital forensics involves techniques for analyzing data in electronic devices and is essential in criminal investigations as evidence in court [1]. There is a subfield of digital forensics that focuses on extracting and analyzing data from mobile devices in a forensically sound manner called mobile forensics [2].

File carving is the process of extracting data from unallocated filesystem space using file type structures [3]. Meanwhile, geotag analysis examines the metadata known as EXIF data embedded in JPEG files to determine the location of media creation, helping the users visualize where their images were captured. However, the process of carving and recovering JPEG files from Android smartphones poses some challenges as data acquisition from Android smartphones is complex due to varying hardware, operating systems, and security features [2]. The widespread use of Android smartphones has resulted in a vast amount of personal data stored in JPEG images, making the accidental loss or deletion of these files a significant concern. Recovering lost JPEG files can be

particularly challenging for non-technical users and extracting valuable geolocation metadata (EXIF data) from these files is often complex. Some free data recovery tools available generally lack integration of geotag analysis, resulting in complex processes requiring more tools. This project has developed an "Android JPEG File Carving with Geotag Analysis" tool that simplifies the recovery of lost JPEG files and the extraction of geolocation metadata. The tool is designed to bridge the gap between advanced data recovery techniques and the needs of normal users, providing a user-friendly solution for recovering and analyzing JPEG data.

The project objectives are to design, develop, and test an “Android JPEG File Carving with Geotag Analysis” tool using an object-oriented approach. The project's scope involves carving standard JPEG files from Android physical images, securely storing files on the user's computer, ensuring data integrity through hash value generation, and performing geotag analysis. The "Android JPEG File Carving with Geotag Analysis" tool holds significant value for general consumers and individuals interested in digital forensics, particularly mobile forensics. By offering a specialized tool that combines Android JPEG file carving with integrated geotag analysis, it enables users to easily recover lost or deleted JPEG files, extract geolocation metadata, and explore Android device investigations in mobile forensics. The integration of "Android JPEG File Carving with Geotag Analysis" tool opens new knowledge and making advanced data recovery techniques accessible to a broader audience.

The report comprises six chapters, starting with an introduction in Chapter 1, covering project background, problem statement, objectives, scope, expected outcome, and project significance. Chapter 2 delves into a literature review, comparing existing and proposed systems. Chapter 3 outlines the development methodology, while Chapter 4 focuses on the design and analysis of the tool. Chapter 5 details the implementation and testing, and Chapter 6 concludes the project, summarizing key findings.

2. Related Work

Digital forensics is a pivotal component of cybersecurity, involving the systematic extraction, preservation, and analysis of electronic evidence. With a focus on legal admissibility, digital forensics aids in investigating cybercrimes, fraud, and other digital incidents. The methodology encompasses a range of activities, from analyzing computer systems to extracting valuable information from digital artifacts [4]. As the most widely used mobile operating system, Android forensics addresses the challenges posed by digital crimes involving Android devices. With considerations like varied device models and file systems, Android forensics requires specialized tools and methodologies. It is a constantly evolving field crucial for extracting and analyzing data from Android devices [5].

2.1 File Carving

File carving stands as a crucial process within digital forensics, enabling the retrieval of files from raw data without relying on file system metadata [6]. This method plays a pivotal role in uncovering digital evidence, particularly in scenarios involving deleted, damaged, or intentionally hidden files. File carving involves identifying file signatures and extracting data lying between them, with early techniques focusing on simple yet effective patterns, such as Header-Footer carving, where specific byte sequences marked the beginning and end of files [7]. In digital forensics, there are three main file carving techniques, which are Header-Footer carving, File Structure-based carving, and Content-based carving, employed to recover lost or fragmented data from storage devices.

2.1.1 Header-Footer Carving

This technique relies on identifying specific markers at a file's beginning (header) and end (footer). In the case of JPEG files, distinctive headers ("FF D8") and footers ("FF D9") mark the beginning and end of files, allowing for accurate extraction [8].

2.1.2 File Structure-based Carving

This technique involves examining the file system structure to identify file boundaries, particularly in cases of fragmentation. For JPEG file extraction, understanding the file system's layout is essential to reconstruct complete files from fragmented data [8].

2.1.3 Content-based Carving

This technique scrutinizes the internal characteristics and content of data to identify file boundaries. In the context of JPEG files, it involves analyzing markers like the Start of Image (SOI) and End of Image (EOI), along with various segments containing image data and metadata. Calculating metadata information over bytes helps reassemble clusters to recover the original file [9].

2.2 Geotag Analysis

Geotagging has gained popularity in digital photography, introducing new possibilities for research in digital forensics. This metadata embedded in JPEG files can provide interesting opportunities for research in the field of digital forensics, bringing together multimedia analysis and computer vision [10]. Geotag analysis involves extracting valuable geographical metadata from digital media, particularly images. It plays a crucial role, containing information such as latitude, longitude, and timestamps. Analysing this metadata aids in reconstructing spatial contexts and establishing timelines for events captured [11]. Despite challenges like intentional manipulation, geotag information can still offer valuable insights into the spatial context and timeline of events.

2.2.1 EXIF metadata

Exchangeable Image Format or EXIF is used for storing a variety of information. It includes the date and time recorded by digital cameras when the photo is taken. Smartphone cameras can also include GPS location. Camera details and settings are also stored in EXIF. The EXIF serves as a standard format using TIFF tags to describe digital images, particularly those in JPEG format. The metadata is organized into JPEG application segments identified by application markers, ranging from binary values 0xFFE0 to 0xFFEF. These segments precede the start of the stream (SOS) segment (0xFFED) containing compressed image data [12].

2.2.2 IPTC metadata

The International Press Telecommunications Council (IPTC) has established a standard for embedding descriptive metadata within digital images. These metadata schemas, along with other emerging standards, provide a uniform format for the creation, processing, and exchange of digital image metadata, facilitating applications in image management, analysis, indexing, and search. The IPTC header standard has become widely adopted for storing and accessing metadata in digital images, with many commercial applications utilizing Adobe's method for inserting and reading IPTC metadata headers. This standard allows users to embed captions, keywords, and text descriptions into their digital images. Additionally, IPTC metadata can include fields for copyright information, enabling photographers to assert their rights and specify usage terms for their images, thus protecting their intellectual property and ensuring proper credit is given for their work. [13]

2.2.3 XMP metadata

The Extensible Metadata Platform (XMP) is an Adobe metadata XML-based schema for storing image metadata. XMP metadata is encoded as XML-formatted text [13]. Unlike EXIF and IPTC metadata, which are primarily focused on describing images, XMP metadata can be used to describe a wide range of digital documents and data sets, including images, videos, audio files, and more. This flexibility makes XMP useful for capturing and maintaining a broad array of metadata, which can enhance the overall context and understanding of digital images in forensic investigations.

2.3 Existing Tools

Digital forensic professionals rely on several existing tools for tasks such as file carving and geotag analysis. FTK Imager, PhotoRec, and PIE (Picture Information Extractor) are some of the most popular applications in this field. Each of these tools has unique capabilities and strengths, providing versatile solutions in the complex landscape of digital forensics.

2.3.1 FTK Imager

FTK Imager is a forensic imaging tool developed by AccessData as a standalone application with capabilities centered around forensic imaging and analysis of storage media. One of the key functionalities of FTK Imager is its ability to create forensic images (exact bit-by-bit copies) of storage media [14]. This process ensures the preservation of original data integrity, which is a critical aspect in forensic investigations. The tool supports multiple forensic image formats, such as raw (dd), SMART, EnCase, and AFF, offering flexibility and compatibility with various forensic tools and environments [14].

2.3.2 PhotoRec

PhotoRec is a software designed specifically for file data recovery. It can recover lost files, including video, documents, and archives from hard disks, CD-ROMs, and digital camera memory. One of the key features of PhotoRec is that it ignores the file system and recovers the underlying data.

2.3.3 PIE (Picture Information Extractor)

PIE (Picture Information Extractor) is a specialized tool used in digital forensics and metadata analysis. It offers a dedicated solution for extracting and examining metadata from image files. PIE is designed to extract and present comprehensive metadata from image files, including EXIF data. It also includes a geotagging feature that allows users to view the photo's position on an interactive map.

2.4 Comparison of Existing Tools

There are various free existing tools available that can perform tasks like file carving and geotag analysis. Table 1 below shows the comparison table between the "Android JPEG File Carving with Geotag Analysis" tool with other existing tools such as FTK Imager, PhotoRec and PIE.

Table 1 *Comparison between the proposed tool with FTK Imager, PhotoRec and PIE Picture Information Extractor*

Tools / Features	FTK Imager	PhotoRec	PIE (Picture Information Extractor)	Android JPEG File Carving with Geotag Analysis Tool
Graphical User Interface	Yes	No	Yes	Yes
Visualization of data	Yes	No	Yes	Yes
Hashing algorithm for verify integrity	Yes	Yes	No	Yes
Wide range of file types support	Yes	Yes	No	No
File carving support	Yes	Yes	No	Yes
Geotag analysis support	No	No	Yes	Yes

Based on Table 1, the "Android JPEG File Carving with Geotag Analysis" tool have all of the key features such as a graphical user interface, data visualization, file carving, and geotag analysis, which are also found in more specialized tools like FTK Imager and PIE. Unlike PhotoRec which only has text-based user interface, the proposed tool offers a user-friendly GUI. "Android JPEG File Carving with Geotag Analysis" tool also ensures data integrity through hashing, like FTK Imager and PhotoRec. However, while FTK Imager and PhotoRec support a wider range of file types, the proposed tool focuses on JPEG files, providing specialized support for geotag analysis not available in absent in FTK Imager and PhotoRec. Overall, this makes the proposed tool a comprehensive option for mobile forensics.

3. Methodology

The development methodology for the "Android JPEG File Carving with Geotag Analysis" tool follows the Agile model, known for its adaptability and iterative approach, encompassing six phases including requirement gathering, design, development, testing, deployment, and feedback [15].

The Requirement Gathering Phase initiates the Agile model, involving various activities such as analyzing user requirements, identifying feasibility, and potential problems for the project. Detailed documentation covering hardware and software requirements is crucial, specifying processor speed, memory capacity, storage preferences, operating system, Python version, required libraries, and other relevant tools. The hardware requirements to ensure optimal tool performance are a quad-core 2.5 GHz processor, a minimum of 8 GB of RAM, running Windows 10 64-bit operating system or above, and at least 256 GB of storage. On the other hand, the software requirements define the tool's operating and development environment, including Python version 3.6 or later, integrated development environments (IDEs) like PyCharm or Visual Studio Code, Android OS version 8.0 or later, and essential libraries like Python Imaging Library (PIL), tkinter, tkintermapview, hashlib, reportlab, etc.

The Design Phase gathered requirements and transformed them into a tangible and user-friendly tool. This phase included creating detailed designs for the tool's architecture, user interface, and functionalities. Various design documents such as algorithm designs, flowcharts, sequence diagrams, interface design wireframes, and test plans were developed to visualize the system and ensure alignment with the defined requirements. The design focused on modularity, reusability, and flexibility, facilitating the implementation of file carving and geotag analysis functionalities. The Development Phase involved the actual coding and integration of the tool using Python, leveraging IDEs like PyCharm and Visual Studio Code. The process was iterative, allowing for gradual building and refining of the tool in small, manageable steps. Key features such as file carving using header-footer techniques, geotag analysis functionalities, and the integration of a hashing algorithm (SHA-256) were developed and tested. The Testing Phase is conducted using functional and user acceptance testing to ensure the tool's reliability and effectiveness. User acceptance testing was conducted via Google Forms by collecting users'

feedback to validate the tool's functionality in real-world scenarios, validating the tool's functionality in real-world scenarios on Android devices.

The Deployment Phase transitioned the tool into a live environment, making it available for end-users. This phase included finalizing the tool, resolving any identified bugs, creating comprehensive user manuals and installation guides, and packaging the tool for user-friendly installation. Lastly, the Feedback Phase concludes the methodology, involving gathering user feedback systematically, analyzing inputs and making necessary adjustments to enhance the tool's functionality and user experience.

4. Analysis and Design

This chapter focuses on the analysis and design of the proposed "Android JPEG File Carving with Geotag Analysis" tool. The chapter covers requirements analysis, algorithms design, Unified Modeling Language (UML) diagrams (Use-Case Diagram, Sequence Diagram, Activity Diagram), test plan design, and user interface design.

4.1 Requirements Analysis

This section provides a comprehensive analysis of user and system requirements for the "Android JPEG File Carving with Geotag Analysis" tool. User requirements, functional requirements, and non-functional requirements are identified and detailed shown in Tables 2, 3 and 4. These requirements serve as the foundation for the subsequent design and development phases.

Table 2 *User Requirements*

No	User Requirements
1	• Users should be able to select directories.
2	• Users should be able to perform JPEG file carving on Android devices' physical images.
3	• Users should be able to pause and resume the file carving process.
4	• Users should be able to cancel the file carving process.
5	• Users should be able to save the carved JPEG file on their computer.
6	• Users should be able to select a JPEG file.
7	• Users should be able to perform geotag analysis on the selected JPEG file.
8	• Users should be able to view JPEG image and geotag information results.
9	• Users should be able to view a visualized map with a pinpoint location based on the geotag analysis result.
10	• Users should be able to generate and save geotag analysis reports.

Table 2 shows the requirements for the "Android JPEG File Carving with Geotag Analysis" tool, detailing the essential functionalities that the tool should provide to meet user needs. These requirements ensure that users can effectively interact with the tool to perform file carving and geotag analysis tasks.

Table 3 *Functional Requirements*

No	Requirements	Description
1	File Carving	• The tool should allow the user to choose to select their input and output directories. • The tool should be able to detect JPEG file signature and carve it. • The tool shall allow the user to pause, resume and cancel the file caring process. • The tool shall allow the user to save the JPEG files to their PC after carving the files.
2	Hash generation	• The tool should be able to generate and add hash value to each of the carved JPEG files before saving it.
3	Geotag Analysis	• The tool should allow the user to select a JPEG file for analysis. • The tool should be able to compare the hash value of the selected image with the hash value stored in the user's computer. • The tool shall display the details of the analyzed geotag information. • The tool should be able to display a preview of the JPEG image and map with the pinpoint location of the geotag.
4	Report generation	• The tool shall allow user to generate and save a PDF report of the geotag analysis result.

Table 3 outlines the functional requirements for the "Android JPEG File Carving with Geotag Analysis" tool which is separated into 4 categories which are file carving, hash generation, geotag analysis and report generation.

Table 4 *Non-Functional Requirements*

No	Requirements	Description
1	Operational	• The tool should be compatible with Android devices' physical images. • The geotag analysis will show a preview image and a map for viewing geolocation information. • The tool runs on Windows 10 or above. • The tool will carve JPEG files from Android devices' physical images. • The tool should be easy to install and deploy.
2	Performance	• The file carving process should be efficient and not overly time-consuming. • The geotag analysis should provide results in a reasonable time frame. • The map should display an accurate pinpoint location based on the analysis result.
3	Security	• The tool will generate and comparing hash value for the carved images to protect its integrity.

Table 4 outlines the non-functional requirements for the "Android JPEG File Carving with Geotag Analysis" tool, which include operational, performance and security requirements.

4.2 Algorithms Design

Algorithm design is a critical process in software development that involves creating step-by-step procedures or formulas to solve specific problems effectively. The two main algorithms of the "Android JPEG File Carving with Geotag Analysis" tool are file carving and geotag analysis are outlined in the Table 5 and Table 6 below. The algorithms help define how JPEG files are identified and extracted from physical images, how their integrity is verified through hashing, and how geotag data is analyzed and reported.

Table 5 *File Carving Algorithm*

Steps	Process
Step 1	Set JPEG_HEADER to the JPEG file header signature.
Step 2	Set JPEG_FOOTER to the JPEG file footer signature.
Step 3	Initialize file_hashes directory to store file hashes.
Step 4	Define input and output directories and other necessary variables.
Step 5	Create necessary directories for output, metadata, and geotagged files.
Step 6	Check if the hash file exists.
Step 7	If it exists, load the hashes into file_hashes.
Step 8	For each file in the selected directory that contain physical images: Read the binary data of the file. Initialize start to 0. While there is more data to process: Find the start of the JPEG file using JPEG_HEADER. If not found, break the loop. Find the end of the JPEG file using JPEG_FOOTER. If not found, break the loop. Extract the JPEG data between start and end. Write the carved JPEG data to the selected output directory. Compute the hash of the carved file and store it in file_hashes. If the file has metadata: Copy the file to the metadata folder. If the file has geotag information: Copy the file to the geotag folder.
Step 9	Save the updated file_hashes to the hash file.

Table 5 shows the algorithm design of the file carving process. The file carving algorithm for the "Android JPEG File Carving with Geotag Analysis" tool is designed to carve JPEG files from physical images of Android devices using Header-Footer carving technique. The process begins with Step 1 and Step 2, where the JPEG file header (JPEG_HEADER) and footer (JPEG_FOOTER) signatures are set. Step 3 initializes a directory to store file hashes, which ensures data integrity throughout the process. In Step 4, the input and output directories, along with other necessary variables, are defined. Step 5 involves creating the necessary subdirectories for output, metadata, and geotagged files. Step 6 checks if a hash file exists, and if it does, Step 7 loads the hashes into the file_hashes directory. In Step 8, the algorithm iterates through each file in the selected directory containing physical images, reads the binary data, and initializes a starting point. Finally, Step 9 saves the updated file_hashes to the hash file, ensuring the integrity of the carved files.

Table 6 *Geotag Analysis Algorithm*

Steps	Process
Step 1	Load stored file hashes from the hash file.
Step 2	Allow the user to select a JPEG file.
Step 3	Display a preview of the selected image.
Step 4	Compute the hash of the selected file.
Step 5	Check if the computed hash matches any stored hash.
Step 6	If no hash value match is found: display a warning
Step 7	Analyze if there is any EXIF data from the image to extract.
Step 8	If EXIF data is found: Extract geotag information and capture time. Convert and format geotag coordinates. Display the location on a visualised map. Display geotag details and capture time in the text container. Else: Display “N/A” for each of the results.
Step 9	Prompt the user if they want to save the PDF report of the analysis report.
Step 11	Create a new PDF document.
Step 12	Set up fonts and styles for the report.
Step 13	Insert the image preview into the PDF.
Step 14	Insert geotag analysis results into the PDF.
Step 15	Insert the report generation timestamp.
Step 16	Save the PDF document.
Step 17	Notify the user of the successful report generation.

Table 6 shows the algorithm table of geotag analysis. The geotag analysis algorithm for the "Android JPEG File Carving with Geotag Analysis" tool. The process begins with Step 1, where stored file hashes are loaded from the hash file. In Step 2, the user selects a JPEG file, and Step 3 displays a preview of the selected image. Step 4 computes the hash of the selected file, and Step 5 checks for a match with stored hashes. If no match is found (Step 6), a warning is displayed. Step 7 checks for EXIF data in the image. If EXIF data is found (Step 8), the algorithm extracts the geotag information and capture time, formats the coordinates, and displays the location on a map along with the details in a text container. If no EXIF data is present, "N/A" is displayed.

Step 9 prompts the user to save a PDF report. If confirmed, Steps 11 to 16 involve creating the PDF document, setting up fonts and styles, inserting the image preview, geotag results, and timestamp, and saving the document. Step 17 notifies the user of the successful report generation.

4.3 Unified Modelling Language (UML)

UML diagrams, including use case diagrams, sequence diagrams, and activity diagrams, are utilized to visually represent different aspects of the “Android JPEG File Carving with Geotag Analysis” tool as shown in Fig 1, 2, 3, 4, and 5 below. These diagrams clearly depict user interactions, system responses, and the flow of activities within the tool.

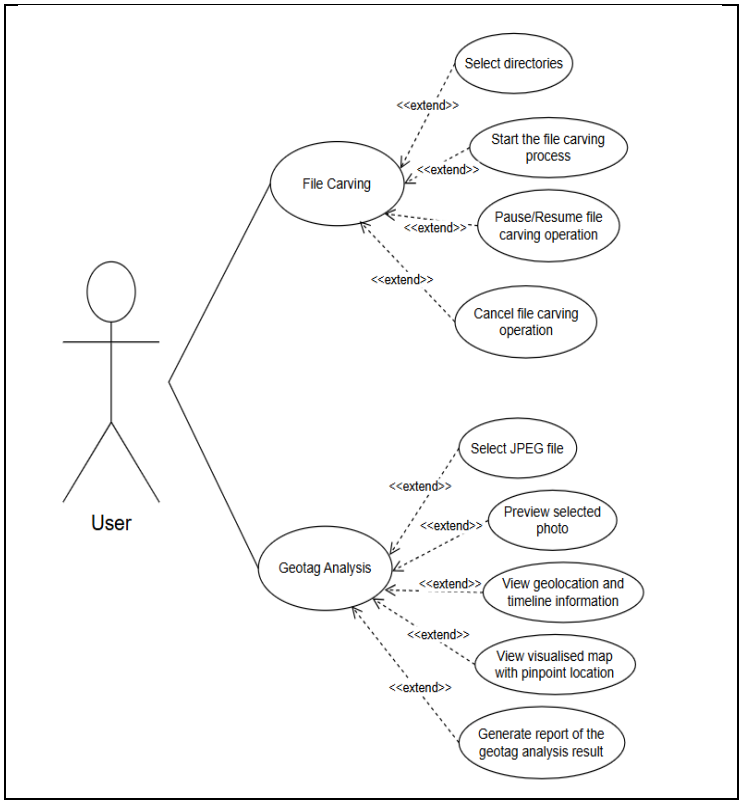


Fig. 1 *Use-Case Diagram*

Fig. 1 shows the use-case diagram for the "Android JPEG File Carving with Geotag Analysis" tool. Based on the figure, there is only one actor, the user, and there are 2 use cases. The first case involved file carving process while the the second case involved geotag analysis.

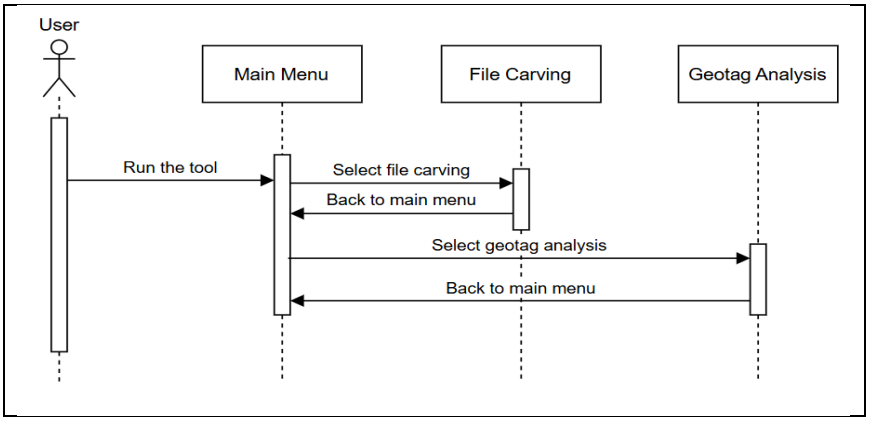


Fig. 2 *Sequence Diagram for Main Menu*

Fig. 2 shows the sequence diagram for the main menu of the tool. User will first enter the main menu when running the tool, they can then choose between file carving or geotag analysis. The file carving module allows users to navigate back to the main menu if they want to use the geotag analysis module and vice versa.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 133-152

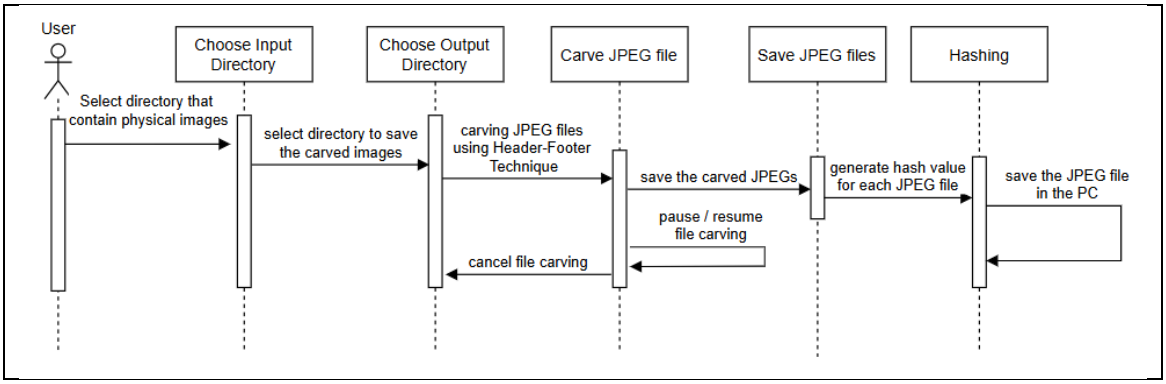


Fig. 3 *Sequence Diagram for File Carving*

Fig.3 shows the sequence diagram for the file carving process of the tool. The user will start by selecting a directory containing physical images and then selecting the directory they desire to save the carved JPEG files. After initiating the file carving process, the tool identifies the JPEG file signature and carved the file. Users can pause, resume and cancel the file carving process at any time, and if the user cancels the process, the user needs to start the process again. After carving the JPEG, it will save the JPEGs to the directory selected by the user. During the carving process, the tool will generate a hash value and assign it to each JPEG file, which the hash values will be stored locally on the user's computer.

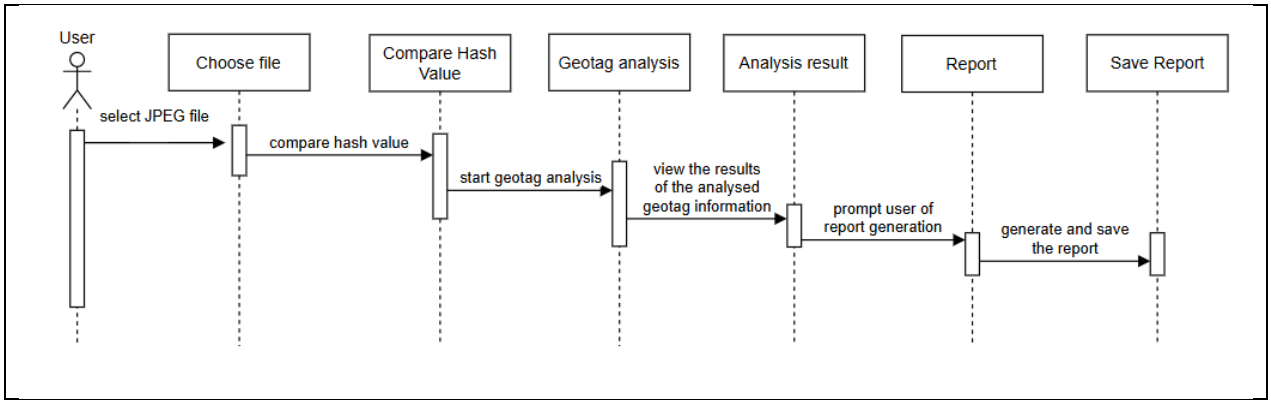


Fig. 4 *Sequence Diagram for Geotag Analysis*

Fig.4 shows the sequence diagram for the geotag analysis. The user will start by selecting the JPEG file, and then the tool will first generate a hash value of the file and compare it to the hash values stored in the user's computer. After that, it will start the geotag analysis process, which analyses the Exif metadata in the JPEG file to extract the geotag information. After finish analyzing the Exif metadata, the user will be able to view the results of the analysed geotag information. Moreover, the tool will prompt user about the report generation. If the user confirms the process, then the tool will automatically generate a PDF report of the analysis result and save it on the user's computer.

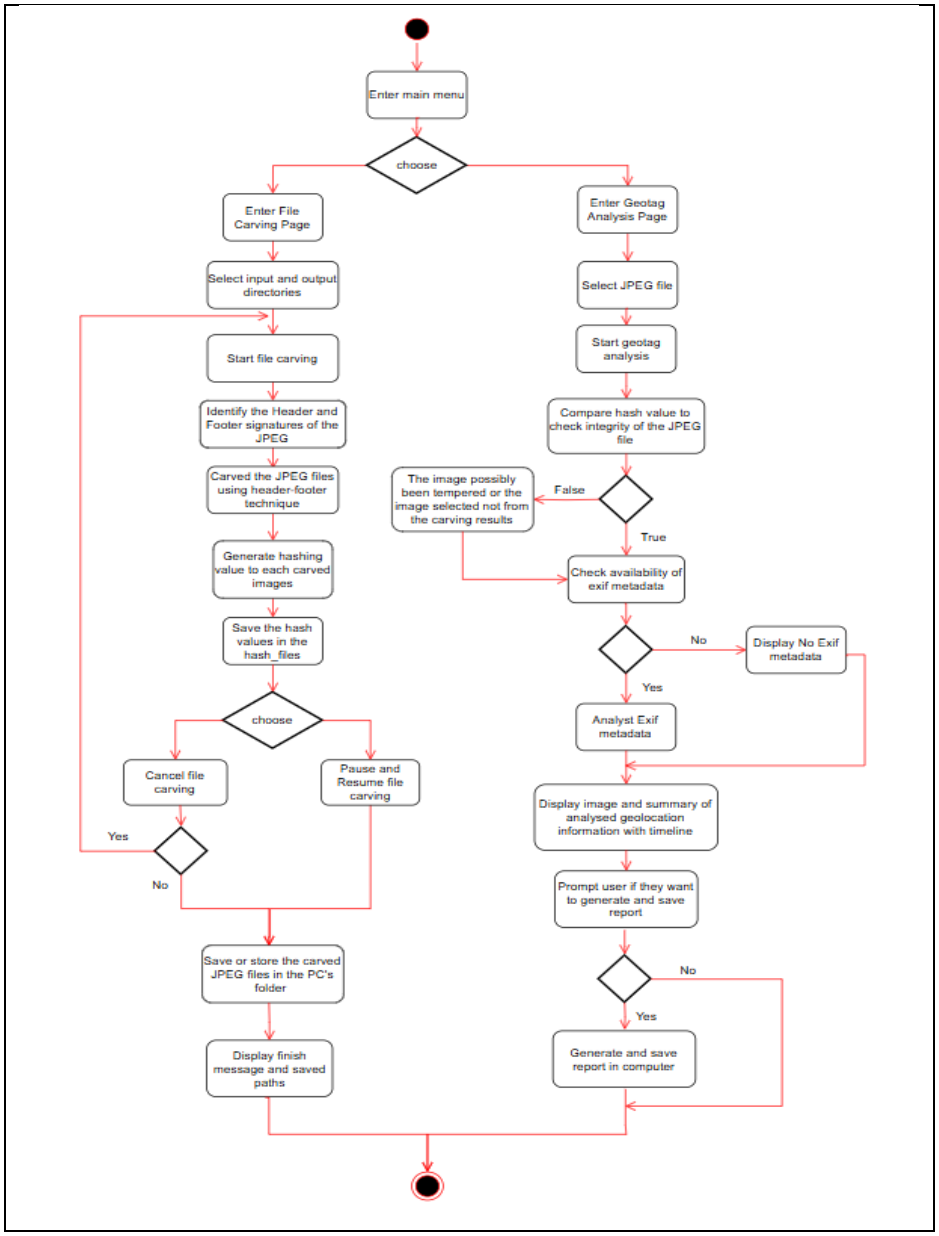


Fig. 5 *Activity Diagram for Android JPEG File Carving with Geotag Analysis tool*

Fig. 5 illustrates the Activity Diagram for the Android JPEG File Carving with the Geotag Analysis tool. Users begin at the main menu, where they select between file carving or geotag analysis. In file carving, users input directories for physical images and carved JPEGs, with the tool scanning for Header and Footer signatures to carve complete JPEG files. Hash values are generated for integrity, and users can pause, resume, or cancel carving. In geotag analysis, users select a JPEG for analysis, with the tool comparing hash values for integrity and checking for EXIF metadata. If available, the tool analyzes metadata, presents geolocation summaries and timelines, previews the image, and offers PDF report generation.

4.4 Test Plan

The Test Plan outlines a structured strategy to ensure the tool's reliability, functionality, and performance. The test plan for "Android JPEG File Carving with Geotag Analysis" tool includes various test cases, such as file carving, hash generation, geotag analysis, and report generation. Each test case specifies the expected result, and actual results will determine whether the test passes or fails.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 133-152

Table 7 *Test Plan*

No	Test Case	Expected Result	Actual Result
1	File carving	• Able to select directories.	Pass/Fail
1	File carving	• The file carving process detects JPEG signatures and carves JPEG files from Android's physical images. • Saving the carved JPEG files to the PC.	Pass/Fail
2	Hash generation	• All carved JPEG images will be attached with a hash value before saving to the PC. • The generated hash values will be stored locally as JSON file on the user's computer.	Pass/Fail
3	Geotag Analysis	• Analyzing a JPEG file displays accurate geolocation information (latitude, longitude, altitude). • Display the capture time and GPS time stamp of the JPEG file. • Display a preview of the image. • Display a map with the pinpoint location of the geotag.	Pass/Fail
4	Report generation	• Generate a PDF report of the geotag analysis result. • Save the report locally on users' computer.	Pass/Fail

Table 7 shows the test plan design for testing all the functionalities of the "Android JPEG File Carving with Geotag Analysis" tool. The testing will cover all the modules and functions available in the tool and will divide the test cases into 4, which are file carving, hash generation, geotag analysis and report generation.

4.5 Interface Design

The Interface Design section focuses on the visual and interactive aspects of the "Android JPEG File Carving with Geotag Analysis" tool. Interface design plays an important part in the development of the tool because it helps describe the early sketch or overview of the tool's interface before the development to ensure that the interface design of the tool is user-friendly. It is also crucial to identify the behaviors and interactions between the user and the tool to enhance the usability and user experience of the "Android JPEG File Carving with Geotag Analysis" tool.

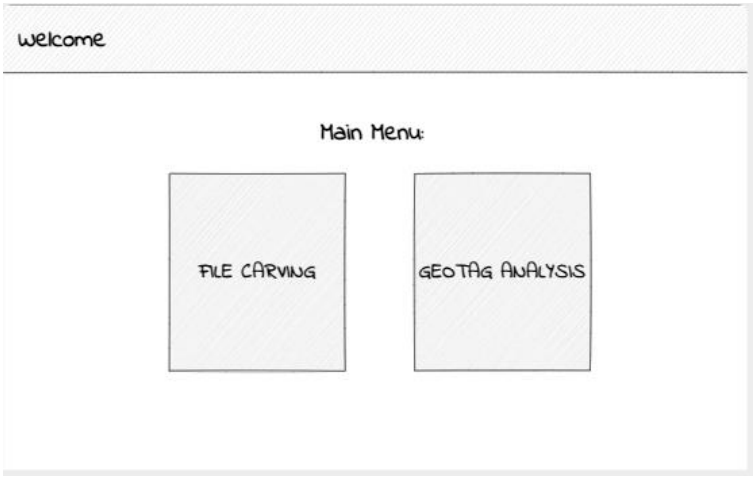
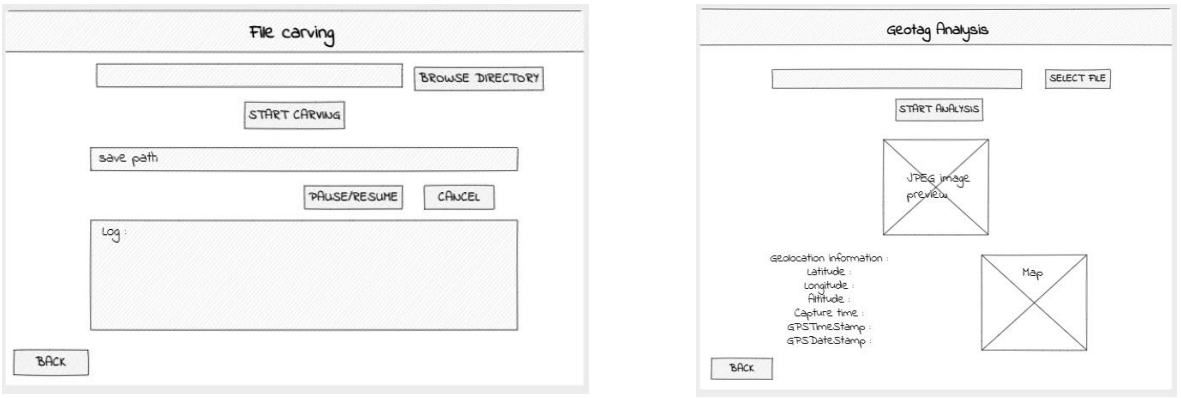


Fig. 6 *Interface design for Main Menu*

Fig.6 shows the main menu interface of the. The main menu mainly has 2 button options for users to choose which activity they want to perform which are file carving and geotag analysis.



(a) **(b)**

Fig. 7 *Interface design for file carving and geotag analysis (a) File Carving Page (b) Geotag Analysis Page*

Fig.7 shows the interface design of file carving page and geotag analysis page. Both the file carving page and geotag analysis page will have a back button on the bottom left corner of the window, allowing the user to navigate back to the main menu. Based on Fig.7 (a), the main contents for the file carving page are the browse button for the user to select the input directory that contains a physical image and the output directory for saving the carved JPEG files. There is also a "Pause/Resume" button that allows users to pause and resume the file carving operation. The "Cancel" button is also available if the user wants to cancel the file carving operation. There is also a textbox located at the bottom of the window that will display a log of the ongoing file carving processes. The geotag analysis page will contain a box that allows the user to drag and drop their desired JPEG file to the tool. textbox located at the bottom of the window.

Based on Fig.7 (b), the geotag analysis page contain a button for user to select their desired JPEG file for the analysis. After selecting their JPEG file, they can click on the start analysis button to perform the geotag analysis on the selected JPEG file. After the process of geotag analysis is finished, the tool will display a preview of the JPEG image, and below the image are the results of the geolocation information of the image. Additionally, the interface will also include a visualized map beside the result textbox that pinpoints the location based on the data to give better visualization to the user.

5. Result and Discussion

This chapter delves into the implementation and testing of the "Android JPEG File Carving with Geotag Analysis" tool. The purpose of the implementation and testing is to ensure the reliability and functionality of the proposed tool in real-world scenarios.

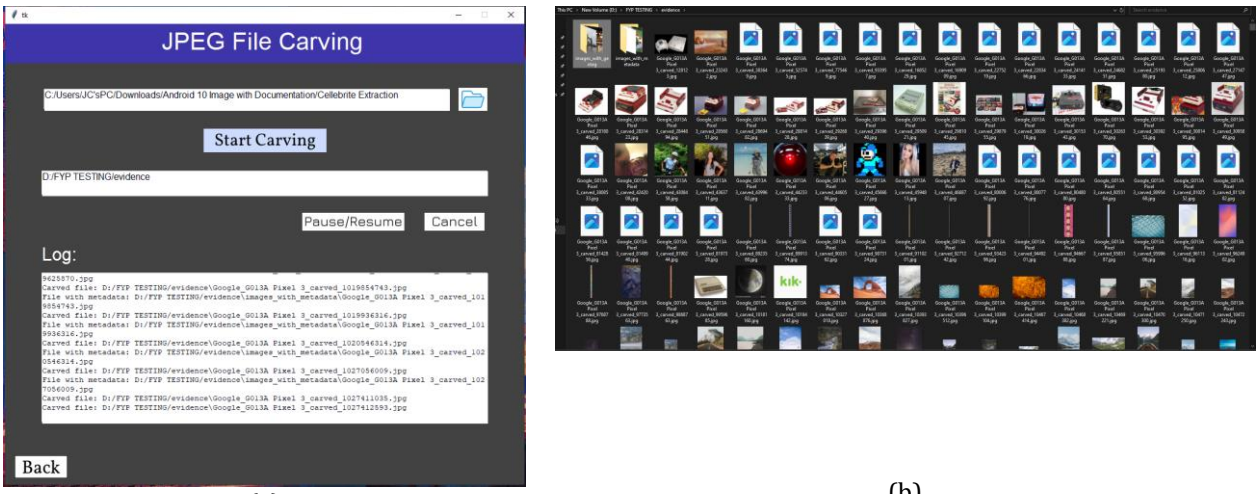
5.1 Implementation of System

This section will discuss the implementation of each interface and module in the "Android JPEG File Carving with Geotag Analysis" tool. The Python programming language was applied in the development of the tool. Implementation is crucial as it involves transferring the designs and requirements of the system into a functional solution. We will delve into the implementations of the interfaces, JPEG file carving, geotag analysis, hash generation and report generation.

(a) **(b)** **(c)**

Fig. 8 *Implementation of interface design (a) Main Menu (b) File Carving (c) Geotag Analysis*

The main menu, illustrated in Fig.8 (a), offers straightforward navigation to file carving and geotag analysis pages. Fig.8 (b) showcases the JPEG file carving interface, featuring browse buttons for input and output directories, along with controls to start, pause/resume, or cancel the carving process. A back button facilitates easy navigation. Lastly, Fig.8 (c) presents the geotag analysis interface, allowing users to select a JPEG image, initiate analysis, view results in a textbox, visualize locations on a map, and return to the main menu.



(a) (b)

Fig. 9 *Implementation of File Carving (a) Output of file carving process (b) Output of saved JPEG files*

Fig.9 (a) shows the output of the file carving process implementation. After the user selects the input and output directories, the input directory will be displayed on the first rectangle textbox, while the output directories will be displayed on the second rectangle textbox. The file carving process will be initiated by the “Start Carving” button. The process of the file carving will be recorded and displayed on the big rectangle textbox. Meanwhile, Fig.9 (b) shows the output of saved JPEG files that were successfully carved out in the output directory selected by the user.

The file carving technique implemented in this tool is Header-Footer technique, which only involves reading the binary data of physical images and searching for JPEG headers ('\xff\xd8\xff') and footers ('\xff\xd9'). The tool identifies the start and end of JPEG files within the binary data of each input file. Then, it extracts this segment of binary data and writes it to a new JPEG file in the output directory once the start and end of a JPEG file are identified. There are also 2 extra folders created in the output directory which are the geotag and metadata folders. If metadata is present, it copies the file to the metadata folder and the file will be copied to the geotag folder if there geotag information found.

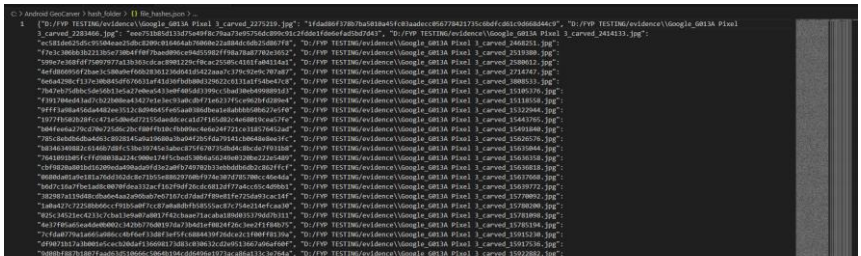


Fig. 10 *Output of generated hash values for each carved JPEG file*

Fig.10 shows the output of the generated hash values for each carved JPEG files. When a file is carved, its contents are read and processed through a hashing algorithm, specifically SHA-256 in this project. Based on the file's content, this algorithm computes a unique fixed-size hash value, typically represented as a hexadecimal number.

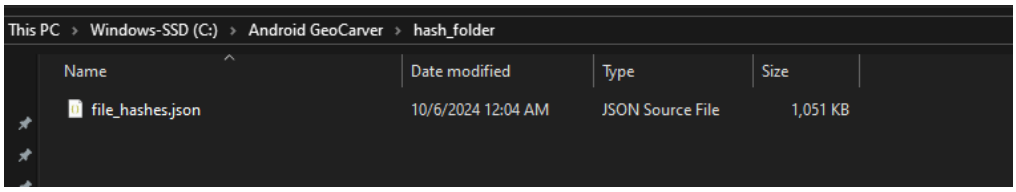


Fig. 11 *Output of saved hash values as JSON file*

Fig.11 shows shows the output of saved hash values in as JSON file. The hash values are then saved as 'file_hashes.json' file in a fixed path initialized and created by the tool.

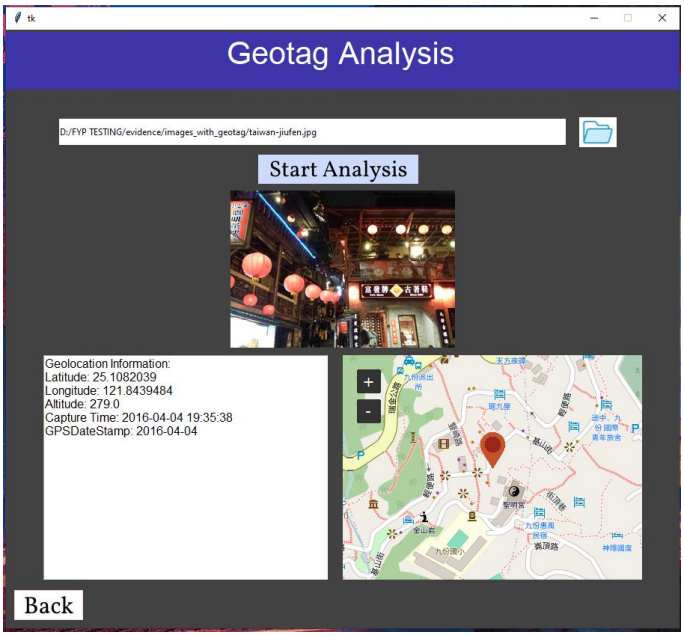


Fig. 12 *Output of geotag analysis process*

Fig.12 shows the output of the implementation in the geotag analysis process. After user select a JPEG and initiate the geotag analysis process, the tool will identify the existence of EXIF and geotag data embedded in the JPEG file. If the data exist, the tool will search for specific tags contain geotag information, including latitude, longitude, altitude, and GPS timestamp. The tool parses these tags to extract geographic coordinates and other relevant data. The tool then interprets the data to make it understandable to the user. This involves converting raw coordinates into human-readable formats and formatting timestamps. The geotag analysis also includes a visual representation of a map. The tool will use the extracted geolocation results to mark the map allowing users to verify the accuracy of the extracted geolocation data by utilizing the Python's library call 'tkintermapview'.

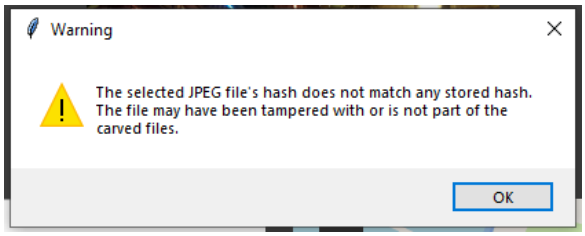


Fig. 13 *Output of warning message of potential tampering of image*

Fig.13 shows the output of the warning message of potential tampering image of the tool due to unmatching hash value when compared to the hash values stored in the user's PC. The tool will compute a cryptographic hash of the file's contents again using a SHA-256 hashing algorithm and then compare it with the hash value saved in 'file_hashes.json'. If the generated hash value doesn't match the hash values stored, there are only 2 possibilities which are the JPEG file has been tampered or the selected JPEG file doesn't belong to the results of carved JPEGs.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 133-152

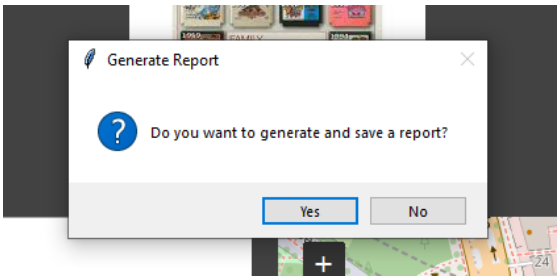


Fig. 14 *Output of message for user's confirmation on report generation*

Fig.14 shows the output of the message for the user's confirmation on report generation. Every time user analyzed a JPEG file, the tool will prompts the user to decide whether they want to generate and save a report.

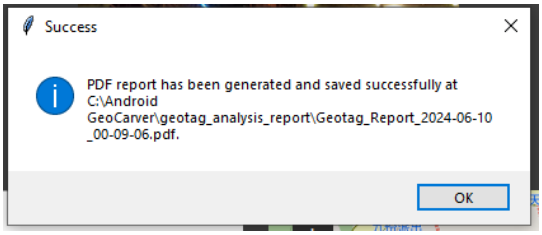


Fig. 15 *The output for the successful message of generated and saved report*

Fig.15 shows the output for the successful message of the generated and saved report. After the user's confirmation of the report generation, there is a dialog notifying the user that the report has been successfully generated and then saved to the created directory.

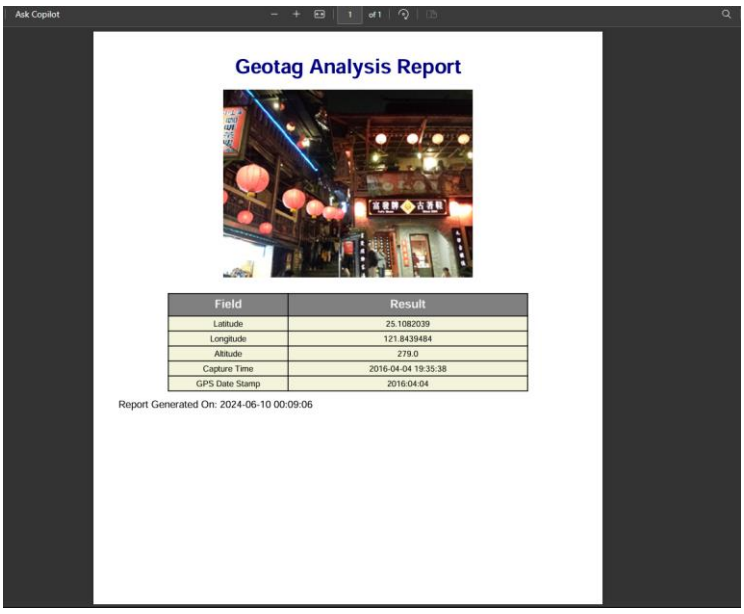


Fig. 16 *Output for generated geotag analysis report*

Fig.16 shows the output of the report in PDF format, which consists of a title, image, geotag analysis result and report generation timestamp as its contents. The geotag analysis result will be displayed in table form to better readability and formality. If the tool didn't identify EXIF or geotag metadata in the selected image. The result fields will be filled with 'N/A' instead.

5.2 System Testing

This section will outline the testing performed, which are functional testing and User Acceptance Testing (UAT). Both testing aimed to evaluate the overall functionality, usability and reliability of "Android JPEG File Carving with Geotag Analysis" tool in real-world scenarios by targeted users.

Table 8 *List of Test Cases*

No.	Test Cases	Description
TEST_100		
1	TEST_100_001	User selects the input and output directories.
2	TEST_100_002	The user starts the file carving process, in which the tool detects JPEG signatures and carves JPEG files from physical images.
3	TEST_100_003	The tool saves the carved JPEG files to their PC.
TEST_200		
1	TEST_200_001	The tool attaches a hash value to each carved JPEG image before saving it to the PC.
2	TEST_200_002	The generated hash values are stored locally as a JSON file on the user's computer.
TEST_300		
1	TEST_300_001	User selects a JPEG file for analysis; the tool displays accurate geolocation information (latitude, longitude, altitude).
2	TEST_300_002	The tool displays the capture time and GPS timestamp of the JPEG file.
3	TEST_300_003	The tool displays a preview of the selected image.
4	TEST_300_004	The tool displays a map with the pinpoint location of the geotag.
TEST_400		
1	TEST_400_001	The tool generates a PDF report of the geotag analysis result.
2	TEST_400_002	The report is saved locally on the user's computer.

Table 8 above shows the list of test cases designed to verify the functionality of the "Android JPEG File Carving with Geotag Analysis" tool. This test cases have been categorized into 4 groups, each focusing on a specific functionality of the tool. The test cases include TEST_100, TEST_200, TEST_300 and TEST_400, which represent the test case for each of the functionalities of the tool, which are file carving, hash generation, geotag analysis, and report generation, respectively.

Table 9 *List of Test Case Result*

No.	Test Cases	Expected Result	Actual Result
TEST_100			
1	TEST_100_001	The tool allows the user to select input and output directories without errors.	As Expected
2	TEST_100_002	The tool detects JPEG signatures and successfully carves JPEG files from the Android's physical images.	As Expected
3	TEST_100_003	The carved JPEG files are successfully saved to the specified directory on the PC.	As Expected
TEST_200			
1	TEST_200_001	All carved JPEG images have a hash value attached before saving to the PC.	As Expected
2	TEST_200_002	The generated hash values are correctly stored in a JSON file on the user's computer.	As Expected
TEST_300			
1	TEST_300_001	The tool accurately displays geolocation information (latitude, longitude, altitude) for the selected JPEG file.	As Expected
2	TEST_300_002	The capture time and GPS timestamp of the JPEG file are correctly displayed.	As Expected
3	TEST_300_003	The selected JPEG image is displayed as a preview.	As Expected
4	TEST_300_004	The map with the pinpoint location of the geotag is correctly displayed.	As Expected

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 133-152

Table 9: (cont)

No.	Test Cases	Expected Result	Actual Result
TEST_400			
1	TEST_400_001	The PDF report of the geotag analysis result is generated without errors.	As Expected
2	TEST_400_002	The PDF report is successfully saved to the specified directory on the user's computer.	As Expected

Table 9 above presents the expected and actual outcomes for each test case outlined in Table 5.1. The expected results specify the correct behavior for each test scenario, such as the successful selection of directories, accurate detection and carving of JPEG files, attachment and storage of hash values, and precise geotag analysis displaying location and timestamp data. Based on the Table 9 results, the testing showed that the tool generally performed as expected, with successful directory selection, accurate JPEG carving, and correct hash generation. The geotag analysis module effectively displayed the necessary geolocation information and map visualizations, while the report generation functionality accurately produced and saved PDF reports. After the tool passed the functional testing, it will undergo user acceptance testing that involve result which will be shown in Fig.17, Fig.18, Fig.19 and Fig.20 below.

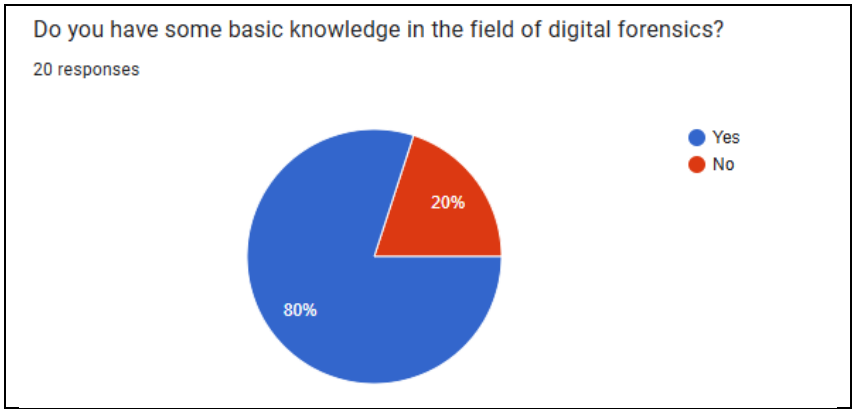


Fig. 17 *User's Digital Forensics Background Knowledge Result*

Fig.17 shows the pie chart that illustrates the results showing that from 20 participants, 80% of the respondents reported having basic knowledge of digital forensics, while 20% did not. This data indicates that most users who participated in the testing possess a fundamental understanding of digital forensics. Even though there are 20% of users without a background in digital forensics, their positive feedback can help indicate that the tool is intuitive and user-friendly, making it accessible even to those without prior expertise in the field.

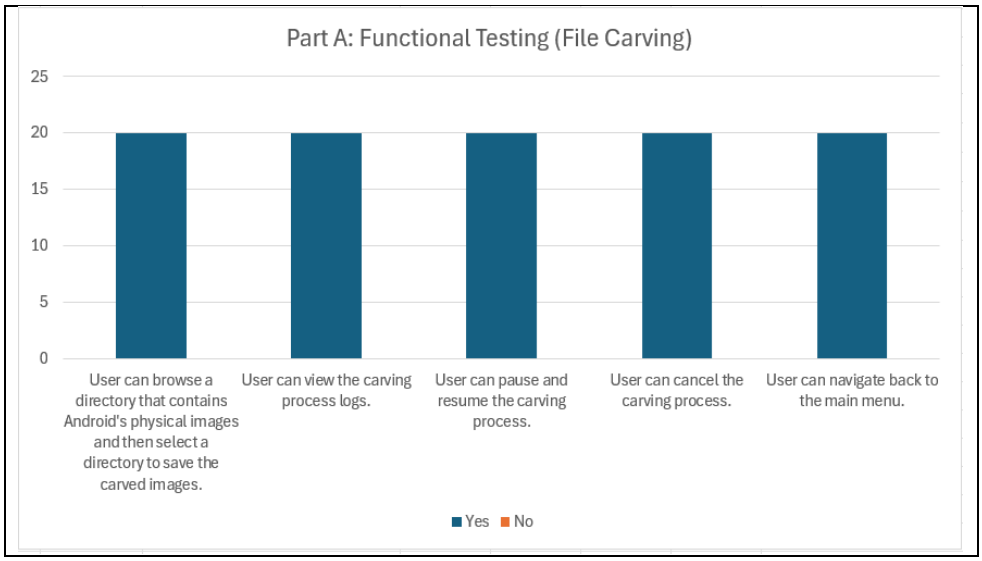


Fig. 18 *User Acceptance Result for Part A in Google Form*

Fig.18 shows the results of user acceptance testing on the file carving functionality of the tool. The results show that all 20 users could successfully browse directories containing Android's physical images and select a directory to save the carved images. This indicates that the directory selection process is intuitive and works flawlessly. Moreover, all users could view the carving process logs, which suggests that the logging functionality effectively provides real-time feedback on the process. The ability to pause and resume the carving process was also confirmed by all users, indicating that these features are functioning correctly. Similarly, the option to cancel the carving process was successfully used by all users and all users can navigate back to main menu page.

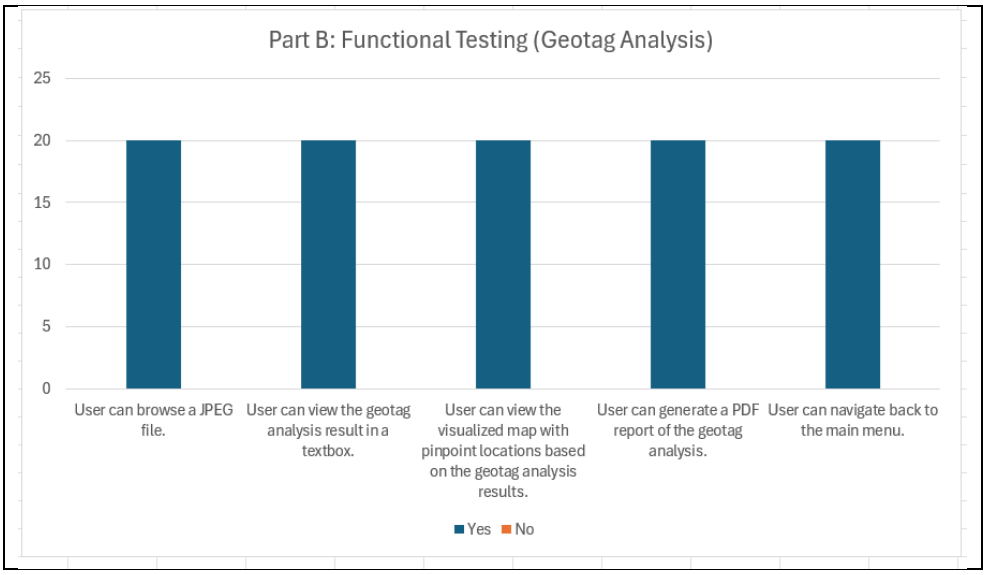


Fig. 19 *User Acceptance Result for Part B in Google Form*

Fig.19 shows the results of user acceptance testing on the file geotag analysis functionality of the tool. The results for geotag analysis functionality is also equally positive as all 20 users successfully browsed and selected a JPEG file, indicating that the file browsing functionality is efficient and easy to use. The geotag analysis results were correctly displayed in a textbox for all users, showing that the tool successfully processes and presents geotag data. Users also reported that the visualized map with pinpoint locations based on the geotag analysis results worked as expected. Furthermore, all users could generate a PDF report of the geotag analysis, which suggests that the report generation feature works as intended. Lastly, the navigation to main menu function also working as smoothly similar to file carving module.

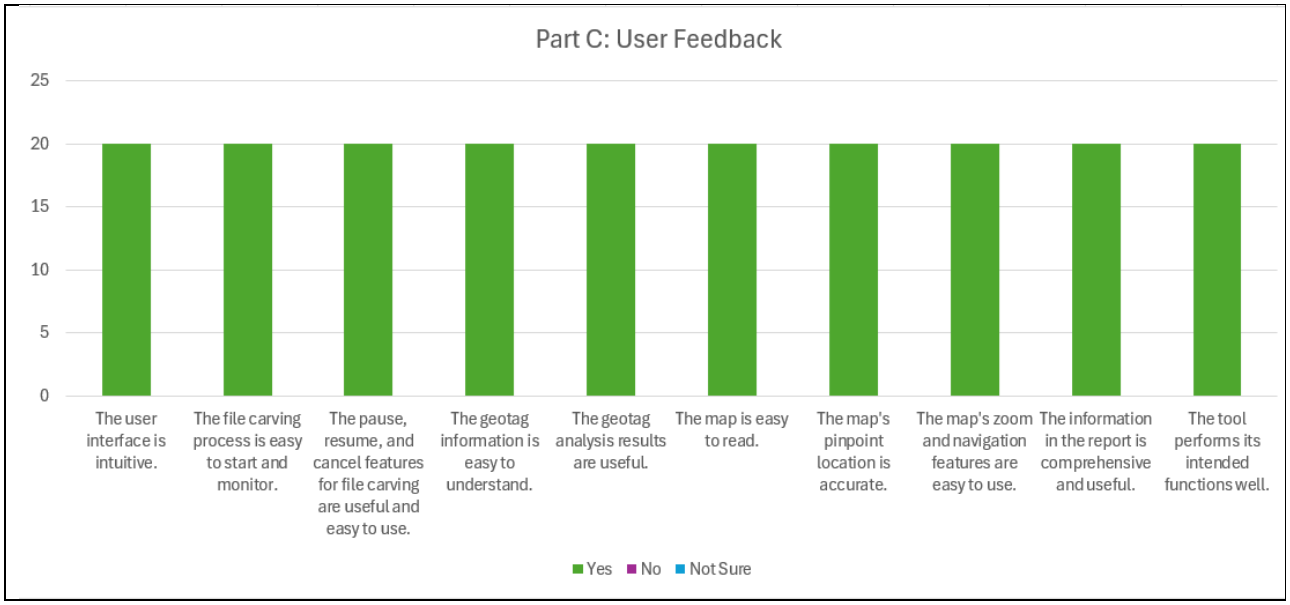


Fig. 20 *User Acceptance Result for Part C in Google Form*

Fig.20 illustrates the results of the tool's overall user experience and usability. The results indicate that all users found the user interface intuitive, which reflects the design's success in making the tool easy to navigate and use. he file carving process was described as easy to start and monitor by all users. The pause, resume, and cancel features for file carving were also deemed useful and easy to use by all users, indicating the effectiveness of these features in enhancing user control over the file carving process. Not only that, the geotag information was easy to understand, and the analysis results were considered useful, which indicates that the tool provides clear and valuable geotag data.

All of the users agreed that the map features, including readability and pinpoint accuracy were good, demonstrating that the tool's geotag visualization is both functional and user-friendly. Lastly, the information in the report was described as comprehensive and useful, and all users agreed that the tool performs its intended functions well. In conclusion, the overall positive feedback confirms that the tool meets its functional requirements and provides a high level of user satisfaction.

6. Conclusion

The "Android JPEG File Carving with Geotag Analysis" tool has been successfully designed, developed, and tested, meeting the project's objectives of enhancing digital investigations in mobile forensics. By integrating file carving techniques with geotag analysis, the tool empowers users to explore and attempt extract JPEG files from Android devices' physical images and analyze geolocation metadata within the images. The tool exhibits modularity, reusability, and flexibility through an object-oriented approach, which makes future enhancements and adaptations possible.

The tool offers several advantages, including a user-friendly interface, file carving capabilities for multiple Android images, informational geolocation and metadata extraction, visualized map representations, integrity verification through hash comparisons, and comprehensive report generation. These features collectively streamline the forensic analysis process, providing users with a robust toolset for examining digital evidence from mobile devices. However, the tool has certain limitations such as the scope of the tool is currently restricted to standard JPEG files, limiting its applicability to other image formats. Additionally, the effectiveness of geotag analysis relies on the presence of EXIF metadata within the JPEG files. There is also some compatibility issues with certain file formats or operating systems that may arise, and the tool cannot handle data fragmentation during the carving process. Furthermore, the simplicity of the generated reports may impact the depth and clarity of forensic investigation reports.

In order to address these limitations and further enhance the tool's capabilities, future implementations could focus on expanding file format support to include PNG, TIFF, and BMP, developing algorithms to handle data fragmentation, and improving reporting functionality with customizable templates and visual aids. These enhancements would broaden the tool's applicability, improve its effectiveness in carving incomplete files, and enhance the clarity of forensic investigation reports.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia for its support..

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Tam Jia Cherng, Nurul Azma Abdullah;* ***data collection:*** *Tam Jia Cherng, Nurul Azma Abdullah;* ***analysis and interpretation of results:****, Tam Jia Cherng, Nurul Azma Abdullah;* ***draft manuscript preparation:*** *Tam Jia Cherng, Nurul Azma Abdullah. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] J. Cai, L. Dawson, G. T. Javan, S. Özsoy, F. C. Quaak, and T. K. Ralebitso-Senior. (2018). "From Experimental Work to Real Crime Scenes and the Courts", In Forensic Ecogenomics, pp. 177-209.
[2] S. Bommisetty, R. Tamma, and H. Mahalik. (2014). Practical mobile forensics. Packt Publishing Ltd.

[3] S. Tahiri. (2016). Mastering mobile forensics. Packt Publishing Ltd.
[4] B. Ciaramitaro. (2013). "Digital forensics explained," CRC Press.
[5] A. Menahil, W. Iqbal, M. Iftikhar, W. Shahid, K. ul Hassan, and S. Rubab. (2021, November). "Forensic Analysis of Social Networking Applications on an Android Smartphone," Wirel Commun Mob Comput, vol. 2021, pp. 1–36, Nov. 2021, doi: 10.1155/2021/5567592.
[6] A. Pal and N. Memon. (2009). "The evolution of file carving," Signal Processing Magazine, IEEE, vol. 26, pp. 59–71, Nov. 2009, doi: 10.1109/MSP.2008.931081.
[7] D. Povar, and V. K. Bhadran. (2011). "Forensic data carving", In International Conference on Digital Forensics and Cyber Crime, pp. 137-148, 2011.
[8] M. I. Cohen. (2007). "Advanced Carving Techniques. Digital Investigation", 4(1-4), pp. 119-128, 2007.
[9] J. Metz, and R. J. Mora, "Analysis of 2006 DFRWS Forensic Carving". DFRWS (2006) Challenge. <http://sandbox.dfrws.org/2006/mora/dfrws2006.pdf>
[10] J. Luo, D. Joshi, J. Yu, and A. Gallagher. (2011). "Geotagging in multimedia and computer vision—a survey," Multimed Tools Appl, vol. 51, pp. 187–211, 2011.J. Luo, D. Joshi, J. Yu, and A. Gallagher, "Geotagging in multimedia and computer vision—a survey," Multimed Tools Appl, vol. 51, pp. 187–211, 2011.
[11] S. Morrissey. (2010). "GPS Analysis," in iOS Forensic Analysis for iPhone, iPad, and iPod touch, Berkeley, CA: Apress, 2010, pp. 227–265. doi: 10.1007/978-1-4302-3343-5_7.
[12] Hamilton E (2004) JPEG File interchange format. Interchange 81:467–490. http://www.dspace.cam.ac.uk/handle/1810/54
[13] J. Tesic. (2005). "Metadata practices for consumer photos," IEEE MultiMedia, vol. 12, no. 3, pp. 86–92, 2005, doi: 10.1109/MMUL.2005.50.
[14] A. Data, "FTK imager." 2020
[15] G. Kumar and P. K. Bhatia. (2012). "Impact of agile methodology on software development process," International Journal of Computer Technology and Electronics Engineering (IJCTEE), vol. 2, no. 4, pp. 46–50, 2012.