Applied Information Technology And Computer Science Vol. 1 No. 1 (2020) 173–180

AITCS

Homepage: http://penerbit.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN : 2-0000

Data Carving Linearly Fragmented JPEG Using File Structured Based Technique

Tan Kin Fei, Nurul Azma Abdullah*

Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, MALAYSIA

DOI: https://doi.org/10.30880/ aitcs.2020.01.01.017
Received 08 November 2020; Accepted 29 November 2020; Available online 30 December 2020

Abstract: Data carving is a robust technique that is used in the digital forensic investigation for data recovery. It is a file recovery technique that extract files based on the file format characteristic specifics and structure in which it can recover data without relying on metadata. Over the years, it has gathered the attention of researcher as it can overcome the shortcomings of the orthodox method of file recovery. However, many of the existing file carving tools were not specifically aimed and designed in consideration of fragmented file and fragmentation of a file can exist in different multitude of scenario. Thus, this paper proposes file carving tool that can recover linearly fragmented JPEG file using structured based technique where this technique utilizes the internal layout of the JPEG file with its elements such as header, footer, and the file size, to carve out non fragmented as well as linearly fragmented JPEG image. The proposed tool is tested and evaluated using Digital Forensic Research Workshop (DFRWS) 2006 challenge dataset where the proposed tool are able to carve out 8 JPEG images that are aligned with the project scope and the remaining unrecovered images are out of project scope where it is bi-fragmented or contained non-JPEG files in the fragmentation. Hence, the result shows that this tool has is capable of carving non-fragmented as well as linearly fragmented JPEG image and the objectives are achieved.

Keywords: Carving, Fragmented, Recovery

1. Introduction

In the modern digital age that is driven by the vast knowledge and advancement in information technology, it is no coincident that computer crime has also been growing exponentially in conjunction to the degree in which computer science has established today. Digital forensics practitioner is responsible for the reconstruction of evidence that can hold up to scrutiny that are reliable to be presented in a court of law [1]. The objective assessment of evidence involved processes of data extraction and preservation as it is often the case that there is a high proclivity in the perpetrator destroying the evidence of the committed crime stored in the multimedia files. In crime cases such as child pornography, forensics expert uses file carving technique and often successfully recover images from the hard disks of the suspect despite the deliberate damages being conducted by the suspect to the filesystem.

*Corresponding author: azma@uthm.edu.my

The orthodox approach to the issue of recovering files is predicated upon the metadata of the file such last access date, are often recoverable [2]. The data recovery method that uses metadata is relatively easy as the metadata points to the clusters allocated to the files. However, in the event where the metadata of the file is unavailable, carving technique is utilized to extract and recover the loss data. File carving is practically effective in extracting structured data from a medium storage, based on the file format specific characteristics present in the filesystem. It is primarily useful in recovering data where the data entries are unavailable or the metadata of the filesystem is deleted, corrupted or more overwritten. Essentially, data carving reconstructs data by examining the header and footer of a file and attempt to reassemble the data in between.

There are numerous ways of recovering data and popular existing tools such as Encase and FTK, are only meant to recover data stored sequentially or contiguously in hard disks based on the metadata. While there are some tools such as SCAPEL, able to recover data without metadata, most of them are unable to recover fragmented files. Thus, this work focuses mainly on recovering linearly fragmented data specifically JPEG file format of images. The objective of this project is to design a data carving tool that can recover non-fragmented and linearly fragmented jpeg, to develop the designed data carving tool, and to test the functionality of the developed tool. Hence, the major contributions are summarized as follows:

1. This paper proposed a data carving tool that can help user to recover corrupted JPEG file in terms of its metadata
2. The data carving tool are capable of carving non-fragmented and linearly fragmented JPEG file.

The rest of the paper is organized as follows: Section II describes the application development methodology. Section III present the result. Finally, Section IV concludes the work and highlights a direction for future research.

2. Methodology and System Design

Object-oriented Software Development (OOSD) model is a pragmatic method that focuses on object by constructing visual models of object in developing a software application. The visual models of object helped to apprehend the problems and create the right documentation that produces well-designed programs. Correct visualization of the models will delineate well-defined specifications and different perspective on the framework and the models must be confirmed to check whether it satisfied the user's prerequisite [3].

2.2 System Requirements Analysis

Requirement analysis is a process that will establish condition required to produce or improvise a product. It also determines the capability and constraint of the proposed system based on its operation and implementation. The requirements are user requirements, functional requirements, non-functional requirements which are shown in Table 1, Table 2, and Table 3, respectively.

Table 1: User requirements

No	User requirement
1	User should be able to upload file to the software tool
2	User should be able view the uploaded file from the software
3	User should be able to convert file to hexadecimal
4	User should be able to start carving the uploaded file
5	User should be able to view report of the file
6	User should be able to download file from the software tool

Table 2: Functional requirements

No	Data	Functionality
1	Read media storage	The system should be able to read mass storage media
2	Upload dataset	The system should allow user to upload file
3	View dataset	The system should be able to let user view the uploaded file
4	Convert file	The system should be able to convert data to hexadecimal
5	Carve file image	The system should be able to carve file image

Table 3: Non-functional requirements

No	Data	Functionality
1	Usability	The application tool should be easy to use
2	Operational	• The application tool should be able to display file information • The application tool should be user-friendly • The application tool should be easily maintained • The application tool is available to Windows 10
3	Performance	• The interaction between user and application should not exceed five seconds

2.3 Use Case Diagram

Use case diagram is the simplest representation of user interaction with the system and this describe how a user uses a system to accomplish a specific goal. It also exhibits the user perspective while interacting with the system. Figure 1 shows the use case model of proposed software tool in which the interactions between user and proposed system.

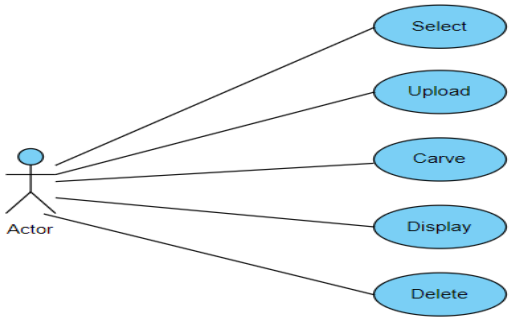

Figure 1: Use case diagram of the proposed tool

2.4 Activity Diagram

Activity diagram is a graphical representation of how the work processes and activity conducted within a system. The purpose of using activity diagram is to delineate the dynamic aspect or behavior of the proposed software tool in the manner of flowcharts. Figure 2 shows the activity diagram of the proposed software tool in which it specifies how the proposed software tool will fulfill its functionality

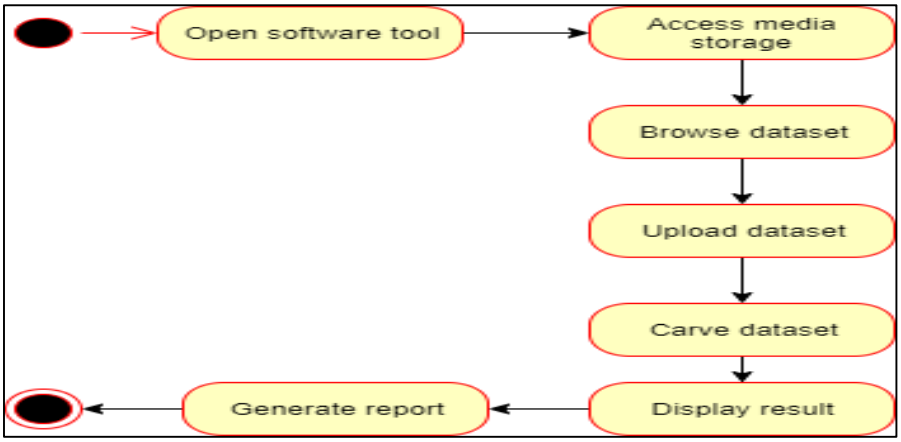

Figure 2: Activity diagram of the proposed tool

2.5 Flowchart

Flowchart is one of the diagrammatic presentations of logic sequence of processes contained in the proposed system. It is especially useful in showing how decisions are made to control events and the flow of operation by presenting the logical relationship between the functions in the system. Figure 3 shows the flowchart for the proposed system.

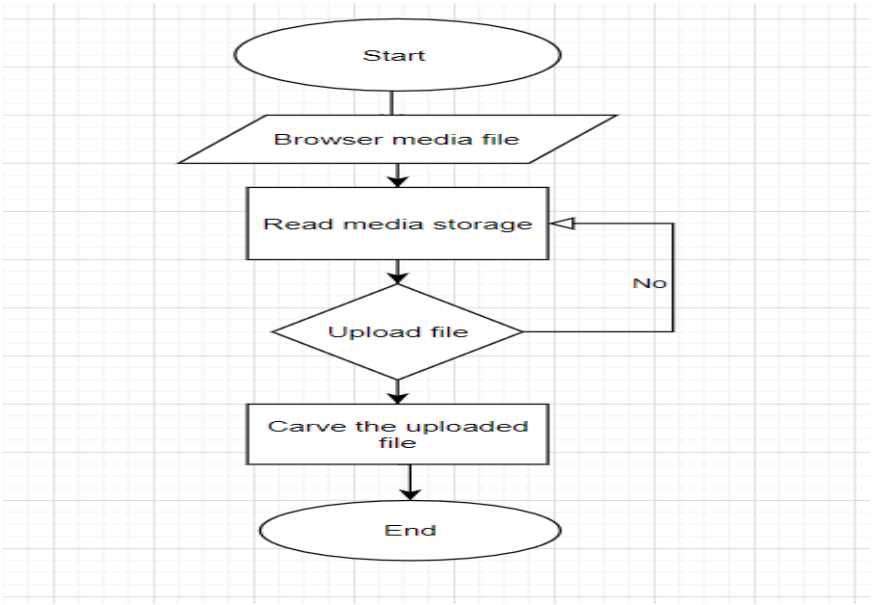

Figure 3: Entity Relationship Diagram of the proposed application

3. Results and Discussion

This section consists of two parts which are implementation and testing of the proposed tool. The analysis and design from the previous section has been implemented and tested. The expected and actual result for the use cases are tabulated and compared.

3.1 System Implementation

JLF carving tool is developed using Python programming language in PyCharm Community Edition. It is written and structured in the manner of object-oriented programming (OOP) paradigm where there

is a usage of class and method, as well as instances of object from a class. Once user open the application tool, user can select file in the file system and proceed to upload and view it in the display widget or start carving processes. Also, there is a delete button for user to delete the uploaded file in display widget if user had uploaded wrong image.

3.2 Detail of datasets

The dataset that is used to test the functionality of the tool is taken from The Digital Forensic Research Workshop (DFRWS) webpage specifically DFRWS 2006 dataset as it contained substantial amount of condition of a JPEG file that aligned with this project scope. DFRWS-2006 dataset is a raw file that contained random data that amounts to the size of 48MB. It has a total of 14 JPEG files with 12 different scenarios and 18 non-JPEG file types that include HTML, Excel, Word, Zip file, and text file with 12 different scenarios. However, this project scope only concerned with JPEG image specifically linearly fragmented and non-fragmented JPEG. Table 6 summarizes the detail and condition of the dataset.

Table 4: Organized datasets with its detail (DFRWS, 2006)

DFRWS 2006 dataset that falls under the project scope	
File	**Scenario**
1b)	One HTML fragmented with a JPEG in between
2d)	One Word file fragmented with a JPEG in between
3a)	One JPEG non-fragmented
3b)	One JPEG non-fragmented, larger than a typical default max file size
3c)	One JPEG non-fragmented, but sector before it has 0xffd8 in the first two bytes
3f)	One JPEG fragmented with random data in between
3g)	One JPEG fragmented with a JPEG in between
3i)	One huge non-fragmented JPEG
DFRWS 2006 dataset that is not within project scope	
File	**Scenario**
3d)	One JPEG fragmented with text in between
3e)	One JPEG fragmented with a Word document in between
3h)	Two JPEGs that are intertwined
3j)	One JPEG fragmented with singe sector in between that starts with 0xffd9

3.3 Functional testing

Functional testing is conducted in of the development of this application. It is to ensure that the design and the functionality of the developed tool are functioning well as intended and fulfill the functional requirement and system specification. This testing phase involved testing of the select function, upload function, carving function, delete function.

3.3.1 Testing of Select function

The select function is tested in this section. If the user clicked select button, a graphical file dialog will pop up that allows user to select file in the filesystem and copy its pathname to the designated label. The result of the testing is shown in table 5.

Table 5: Test for select function

Test case	Expected result	Actual result
Click select button	File dialog appears	File dialog appears
Select file in the file dialog	Pathname of the selected file shown on the label	Pathname of the selected file shown on the label

3.3.2 Testing of Carve function

The carve function is tested in this section. The set of extracted JPEG images used for this testing are from DFRWS 2006 challenge dataset. As previously discussed, the total number of JPEG image contained in the dataset are 14 but only 8 falls under the scope of this project. JLF carving tool able to extract 8 out of 8 JPEG image in the dataset. Also, the carve function is tested for error when user had never selected any file but click carve button. Table 6 shows the result of test that is conducted and figure 6 in Appendix A shows the image that is recovered by the tool.

Table 6: Test of Carve function

Test Case	Expected result	Actual result
Extract Non fragmented JPEG image	Extract viewable image and output in the file system	Extract viewable image and output in the file system
Extract linearly fragmented JPEG image	Extract viewable image and output in the file system	Extract viewable image and output in the file system
Extracted image is saved in the filesystem	Output is saved in the same directory as the selected file	Output is saved in the same directory as the selected file
Click upload button without selecting a file	Error message appears to remind user	Error message appears to remind user

3.3.3 Testing of Upload function

The upload function is tested in this section. The upload button should be able to upload file image to the display widget once user has selected a file and upload it. If user have not selected a file but click upload button, an error message will pop up to remind user to first select a file. Also, if user have already uploaded the image file and click upload again, an error message will pop up to remind user that the file has already been uploaded, all of which is tabulated in Table 7 below.

Table 7: Test for upload function

Test Case	Expected result	Actual result
Click upload button after selection of image file	Image file will be uploaded on the display widget	Image file will be uploaded on the display widget
Click upload button without selecting a file	Error message appears to remind user	Error message appears to remind user
Click upload button twice	Error message appears to remind user	Error message appears to remind user

3.3.4 Testing of Delete function

The delete function is tested in this section. The delete button should be able to delete uploaded image file in the display widget after clicking the radio button that points to that image file. If user have not selected a radio button and click delete button, an error message will pop up to remind user. Hence, the test cases for this function are click delete button after selected an image file, click delete button without selecting an image file, and click delete button more than once as shown in Table 8.

Table 8: Test for delete function

Test Case	Expected result	Actual result
Click delete button after selected an image file	The selected image file will be deleted from the display widget	The selected image file will be deleted on the display widget
Click delete button without selecting an image file	Error message appears to remind user	Error message appears to remind user
Click upload button twice	Error message appears to remind user	Error message appears to remind user

4. Conclusion

In conclusion, the proposed tool has been developed and tested until it reaches its expected result. This tool can help user to extract non-fragmented and sequentially fragmented JPEG image. To recap the objectives stated in section 1, the first objective is to design a data carving tool that can recover non-fragmented as well as linearly fragmented JPEG file. Hence, the first objective has been achieved as the file carving algorithm has been designed and implemented in a tool that is working and meet the expected result. The second stated objective was to develop a data carving tool that can extract non-fragmented and linearly fragmented JPEG file. Therefore, this project also achieved the second objective as this tool had successfully recovered non-fragmented and linearly fragmented JPEG file from DFRWS 2006 datasets as shown in the previous section. The last stated objective was to test the functionality of the tool. This objective has been achieved as every functionality of the tool are tested and the result are tabulated as shown in the previous section. All the clickable buttons in the tool are not only tested for their intended functionality, but also for any error for example, by clicking it twice or upload and delete files without first selecting one. The test result has shown that this tool has met all of its expected output or response.

The proposed tool has several drawbacks. It has relatively narrow scope that it could not recover non-sequential fragmented JPEG image files and lack of proper validation method to identify unwanted cluster and remove error in the decoding process, which be included and improvised in the future work.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this study.

Appendix A

Figure 4: GUI of JLF Carving Tool

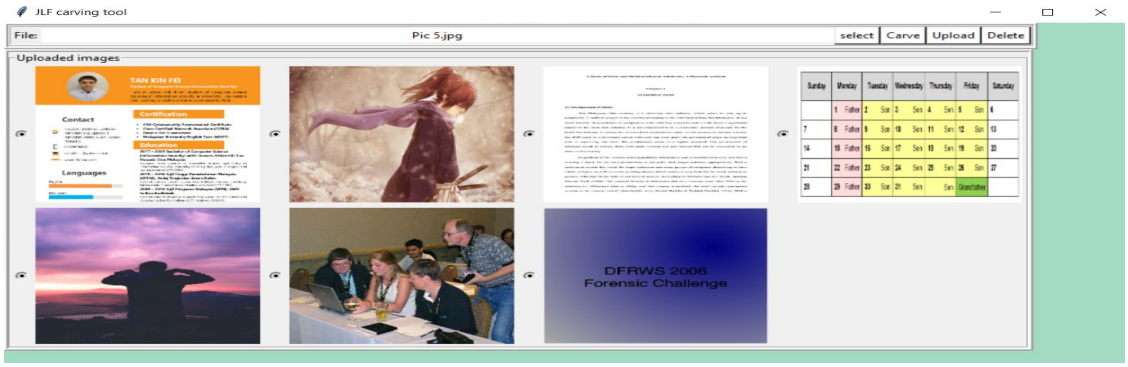

Figure 5: Display Widget of JLF Carving Tool

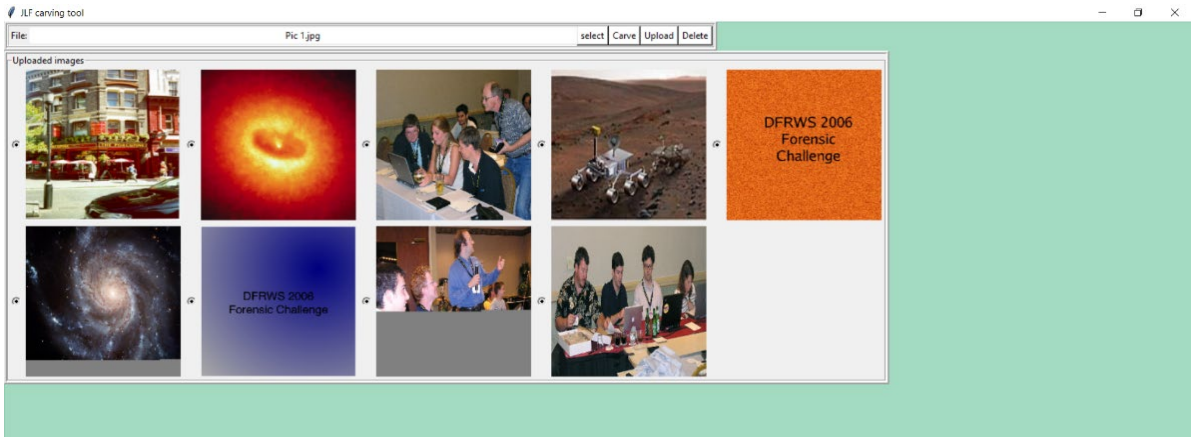

Figure 6: The extracted JPEG image related to the project scope

References

[1] C. Leacock. “Search and seizure of digital evidence in criminal proceedings”. Digital Evidence & Elec. Signature Law Review, vol 5, pp 221-225. 2008.

[2] A. Pal & N. Memon. “The evolution of file carving”. IEEE Signal Processing Magazine, vol 26 Issue 2, pp 59–71. https://doi.org/10.1109/MSP.2008.931081 2009

[3] Y. Singh & R. Malhotra. “Object-oriented software engineering”. PHI Learning Pvt. Ltd. 2012.