Martin Bruchon
Andrew Murphy
31-12-2023

Topic 4: Metadata Scrapper User Guide

Introduction

Our project's objective is to offer a tool that scans and extracts metadata from files present within a specified folder.

This utility is able to handle a diverse range of file formats including audio, video, images, PDFs, and newer Office documents.

The primary purpose of this tool is to simplify the process of metadata extraction from various file types stored in a designated folder. This capability is beneficial for users who need to manage and analyze large volumes of digital files, whether for research, OSINT, or archival purposes.

The tool can store the file into a pickle and csv, so the output can later be used for future analysis.

Getting Started

System Requirements and Installation Instructions

To use this program, ensure that your system has Python 3 installed. We tested on Python 3.8, 3.10, and 3.12. Other versions earlier than 3.8 may have compatibility issues.

The dependencies are listed in the file *requirements.txt*. It is recommended to work within a virtual environment when setting up a new project.

To install the dependencies:

- Open your terminal or command prompt.
- Navigate to the directory containing requirements.txt.
- Run the following command:
  ```
  pip install -r requirements.txt
  ```
- This will install all necessary libraries and packages listed in *requirements.txt*.

Opening the Program

Open your terminal or command prompt and navigate to the folder where *main.py* is located. Then, run the script using Python 3 by typing:

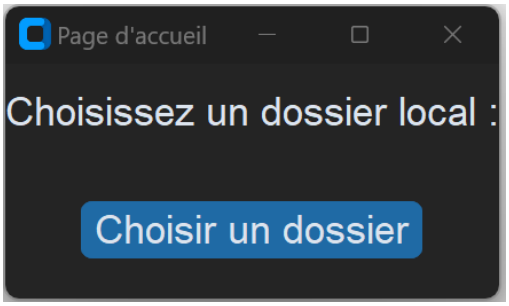
```
python main.py
```

A prompt asking to select a folder will be visible.

Usage Instructions

Begin the program, as described above, by launching the *main.py* program. After this, all options are managed through the GUI. Below are a list of features and how to use them.

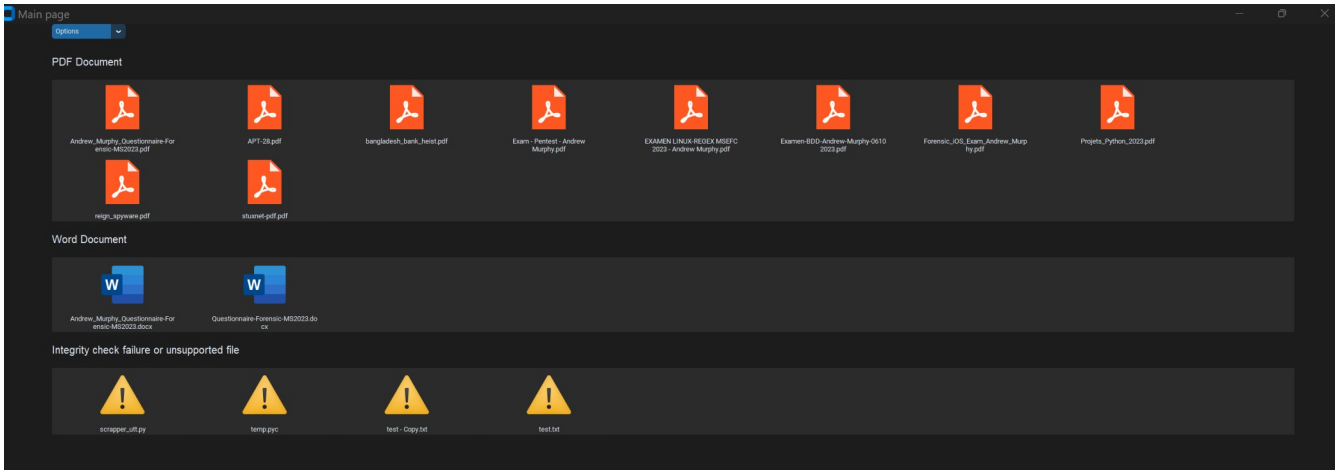
Selecting a Folder



When starting the program, you will be prompted to select a folder. All files within this folder (non-recursively) will be scanned.

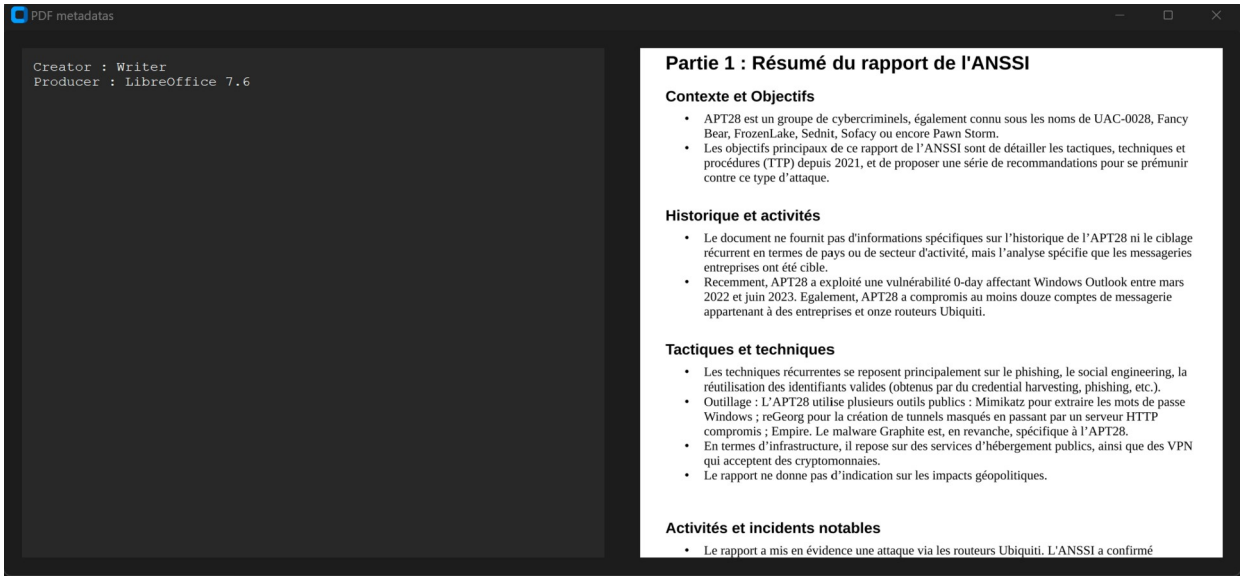
Main page

After the scan is complete, you will see the main window.

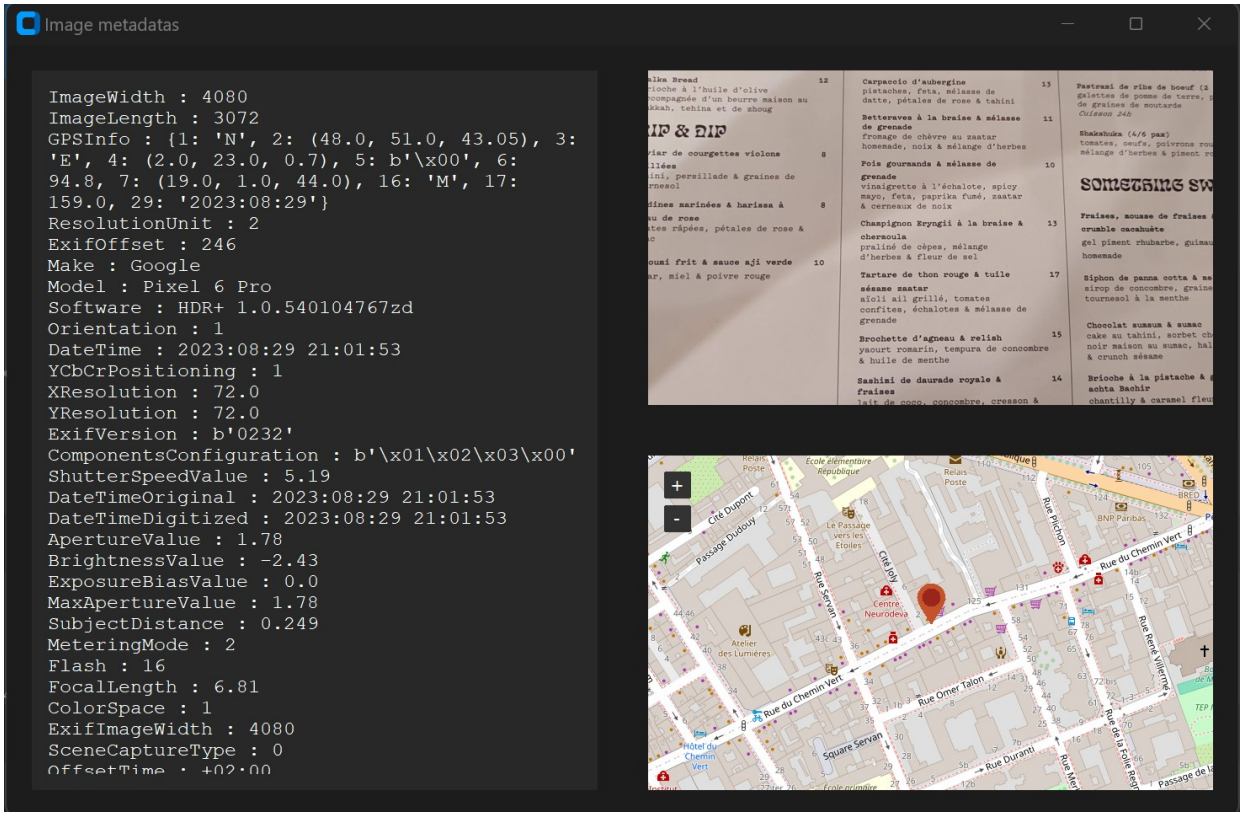


The main window shows a list of all files, organized by file type. Unsupported file types are listed at the bottom.

Single-click on any file to open up more information on that file. A new window will appear with information on the specific file.

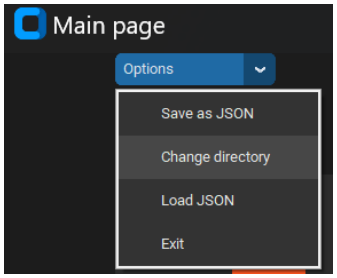


If metadata is present, it will be shown on the left panel. If the object is an image, a pdf, or a video, a preview will be shown on the top-right. And, if any geolocation metadata is present, a map will be shown on the bottom-right.

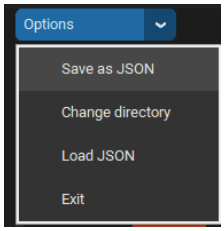


Changing folder

The folder being scanned can be changed at any time by selecting the “Change Directory” button.



Saving Results



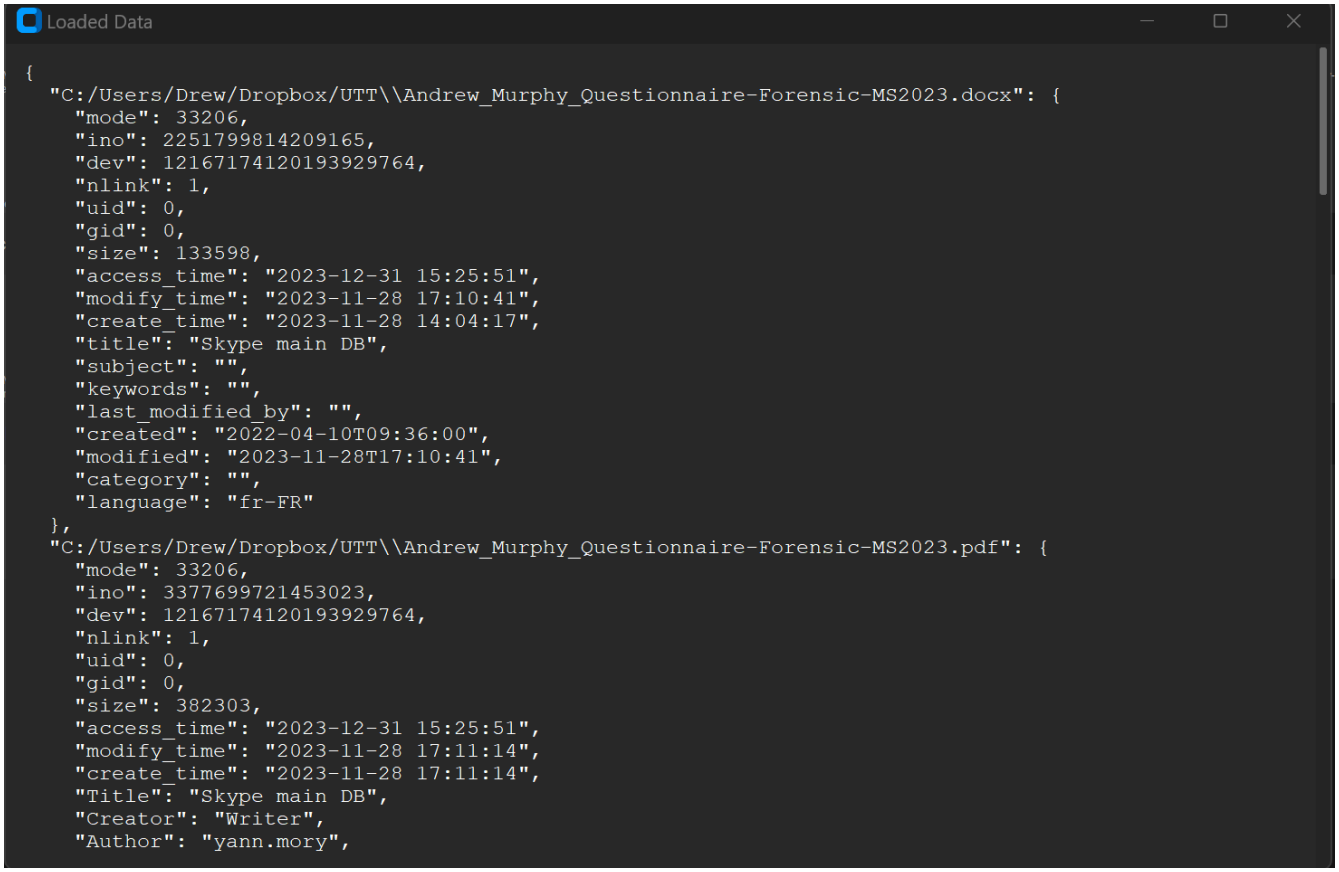
The summary of all files can be saved by clicking the button Save to JSON. The file will be output to the current directory, with the current timestamp appended to the end.

Loading Results

After saving results, you can load them into the program to see the raw output. This will display the json output of each file into a new window.

First, select “Load JSON” from the options bar. Then, navigate to a folder with previously saved JSON output.

The saved information should be printed in a user-friendly format:



List of current supported file formats

Currently, support is limited to the below file types:

Audio Files

- MP3, WAV, and other common audio formats.
- Metadata Extracted: Includes duration, bitrate, artist, album name, and extra information if present in the file.

Office Documents after 2007

- Supported Formats: Microsoft Office formats like DOCX, XLSX, PPTX.
- Metadata Extracted: Author, creation date, last modified date, document title, etc.

PDF Files

- Supported Formats: Standard PDF documents.
- Metadata Extracted: Author, creation and modification dates, document title, number of pages, and more if present in the file.

Image Files

- Supported Formats: JPEG, PNG, and other common image formats. GIF is currently not supported
- Metadata Extracted: Dimensions, creation date, camera settings (if available), file size, etc.

Video Files

- Supported Formats: MP4, AVI, MOV, etc.
- Metadata Extracted: Duration, resolution, codec, frame rate, and other information if present.

For file types not in the above list (txt, word documents before 2007), the files will be grouped into the unsupported format list on the main page.

Program Components

The metadata extractor is made up of modules in different python files.

main.py – Main Program File

main.py serves as the entry point to the application. It calls out to folder_selection.py, which then scans a folder for files and metadata.

folder_selection.py – Folder Scan Selection

folder_selection.py presents the user interface for folder selection and initiates the metadata scanning process. This script integrates various modules to facilitate the analysis of different file types within the selected folder.

audio_info.py - Audio File Metadata

Handles the extraction of metadata from audio files. It processes formats like MP3 and WAV, retrieving details such as duration, bitrate, and artist information.

office_info.py - Office Document Metadata

Focuses on scraping metadata from office documents, including Word, Excel, and PowerPoint files. It extracts information like author, creation date, and last modified date.

pdf_info.py - PDF File Metadata

Dedicated to extracting metadata from PDF files. This includes authorship, creation and modification dates, and other document-specific information.

image_info.py - Image File Metadata

Extracts metadata from various image file formats, including JPEG, PNG, and GIF. It retrieves details like dimensions, creation date, and camera settings (if available).

video_info.py - Video File Metadata

Targets metadata extraction from video files. It supports various formats like MP4 and AVI, gathering data such as duration, resolution, and codec information.

scan_folder.py - Folder Scanning

Manages the scanning of the selected folder. It identifies file types and delegates the extraction process to the respective modules.

metadata_functions.py - Metadata Handling Functions

Contains a collection of functions used across different modules to process and organize metadata. Functions from this file are imported into the individual *info* modules.

home_page.py - Main Interface Component

Responsible for rendering the main user interface of the program. It integrates user inputs and displays results.

Limitations and Next Steps

During development, some choices were made to avoid errors and increase compatibility.

- The full list of metadata is not shown or stored for all files. For many file types, this is extraneous data (pdf files in particular) of no particular importance. This resulted in very large JSON file sizes, and sometimes file write errors. To avoid this problem, we show and store full metadata for images, but only a reduced set of metadata, hand-selected for relevance, for all other file types.
- Window scrolling and zooming was problematic between linux and windows systems. We attempted to make a program that was easy to use in both environments, but there are still some issues with the GUI.
- Export to JSON was chosen because it the most simple to write and then load within Python. We considered CSV and Pickle, but CSV is not easily read in unless the data columns are known, and Pickle has known security issues in python.

In addition, we identified several axes improvements that could be addressed in the future.

pyexiftool library

This would allow extraction of metadata from a much larger variety of files, including txt and older Microsoft Office documents. However, it is only a wrapper, and thus it is dependent on installing *exiftool* within the local environment.

Due to the nature of this project, we wanted to have as few dependencies as possible to avoid complications when installing, so we decided not to move forward with this option.

Additional deployment on Macintosh, mobile, etc.

We did not have access to an Apple computer, so our program was not tested in this environment. However, it was tested in both a Linux and Windows environment.

Given the simplicity of the program, it would also be possible to port this as an android or iOS app. However, this was outside the scope of the project.

User Interface Improvements

We would want to gather feedback on the user experience of our program, in order to improve the GUI. Given time limitations, this step was not done.

About the Team

Martin BRUCHON is a french recently graduated generalist engineer in pursuit of studies in the Mastère Forensics and Cybersecurity at UTT. His main contributions were in the GUI’s development and the management of the extracted metadata.

Andrew MURPHY is a former data scientist at BNP, who is taking the Mastère Forensics and Cybersecurity with the UTT to migrate into a new field. His main contributions were in the core metadata extraction functions and user testing.