SINGAPORE INSTITUTE OF TECHNOLOGY

ICT2201 Digital Forensics

NCMF User Manual

AY2021/2022, Trimester 1

Contents

1. Introduction

This document is a detailed user manual and guide on how to use NCMF's Audio/Image Analyzer.

2. Prerequisites

Required software:

- Python (at least v3.9)

Required Python libraries:

- pydub
- SpeechRecognition
- opencv-python
- Pillow
- pytesseract

Required files to install:

- FFMPEG (Download the latest version from https://www.gyan.dev/ffmpeg/builds/)
- Tesseract (Download the latest version from https://github.com/UB-Mannheim/tesseract/wiki)

3. Set up guide

1. After FFMPEG and Tesseract installation, update your System Environment Variables as shown in Fig 1
   a. Update System Variables' Path to include FFMPEG's bin folder file path and Tesseract installation path as shown in Fig 2
   b. Restart the computer
2. Update tesseract variable with Tesseract.exe's filepath in software code as shown in Fig 3 at line 16 of ncmf.py (Note that the filepath may vary depending on where Tesseract has been installed in your computer)
3. Run the following commands to install necessary libraries and packages
   a. pip install SpeechRecognition
   b. pip install pydub
   c. pip install opencv-pythonz
   d. pip install pytesseract
   e. pip install Pillow

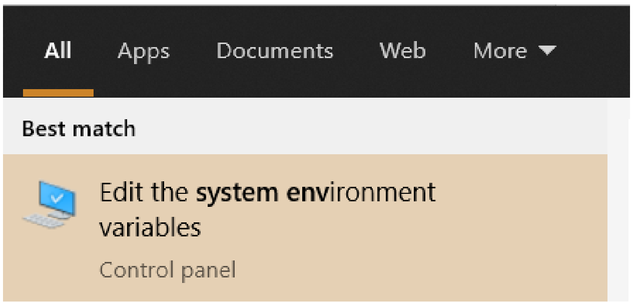
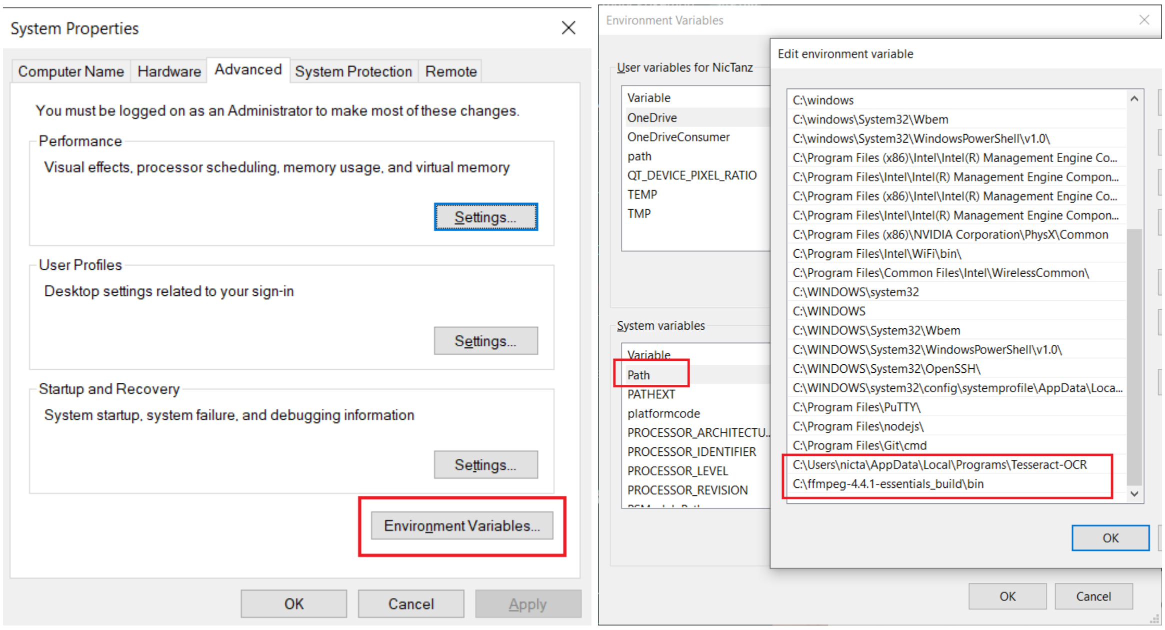


Fig 1



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

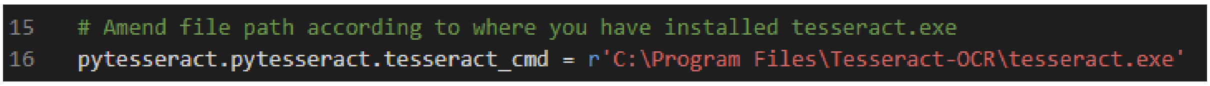
System startup, system failure, and debugging information

Variable
OneDrive
OneDriveConsumer
path
QT_DEVICE_PIXEL_RATIO
TEMP
TMP

Variable
Path
PATHEXT
platformcode
PROCESSOR_ARCHITECTU...
PROCESSOR_IDENTIFIER
PROCESSOR_LEVEL
PROCESSOR_REVISION

C:\windows
C:\windows\System32\Wbem
C:\windows\System32\WindowsPowerShell\v1.0\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Co...
C:\Program Files\Intel\Intel(R) Management Engine Compon...
C:\Program Files (x86)\Intel\Intel(R) Management Engine Co...
C:\Program Files\Intel\Intel(R) Management Engine Compon...
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\WINDOWS\system32\config\systemprofile\AppData\Loca...
C:\Program Files\PuTTY\
C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\Users\nicta\AppData\Local\Programs\Tesseract-OCR
C:\ffmpeg-4.4.1-essentials_build\bin

Fig 2



```
# Amend file path according to where you have installed tesseract.exe
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
```

Fig 3

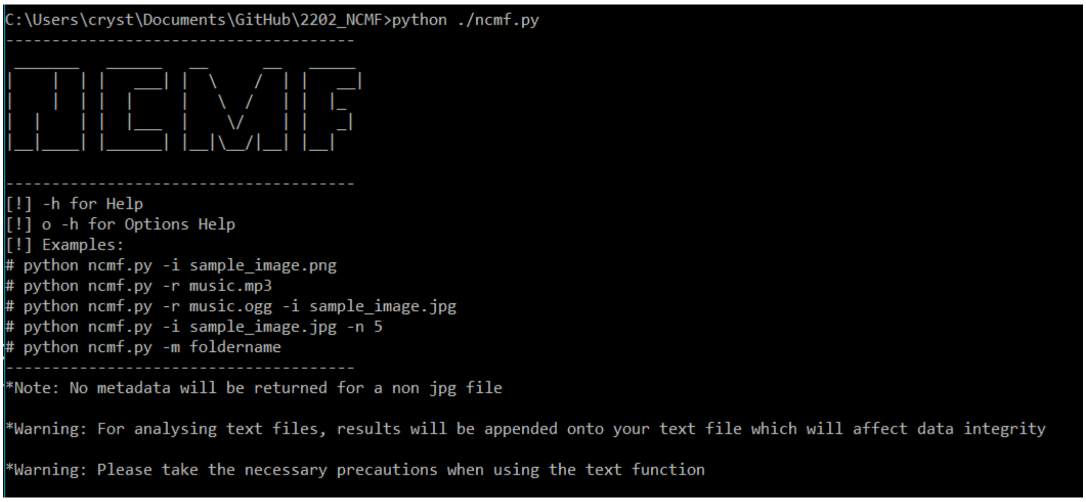
4. Running NCMF

Start Page

To run NCMF, enter the following command in your terminal:

- python ncmf.py

The NCFM's logo, help menu and some examples on how to use NCMF should be displayed.



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py
--------------------------------------
 ______  ______  __     __  _____
|   |  | |  ___| | \   /  | |  __|
|   |  | | |     |  \ /   | | |_
|  |   | | |___  |   V    | |  _|
|__|___| |_____| |_|\__/|_| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function
```

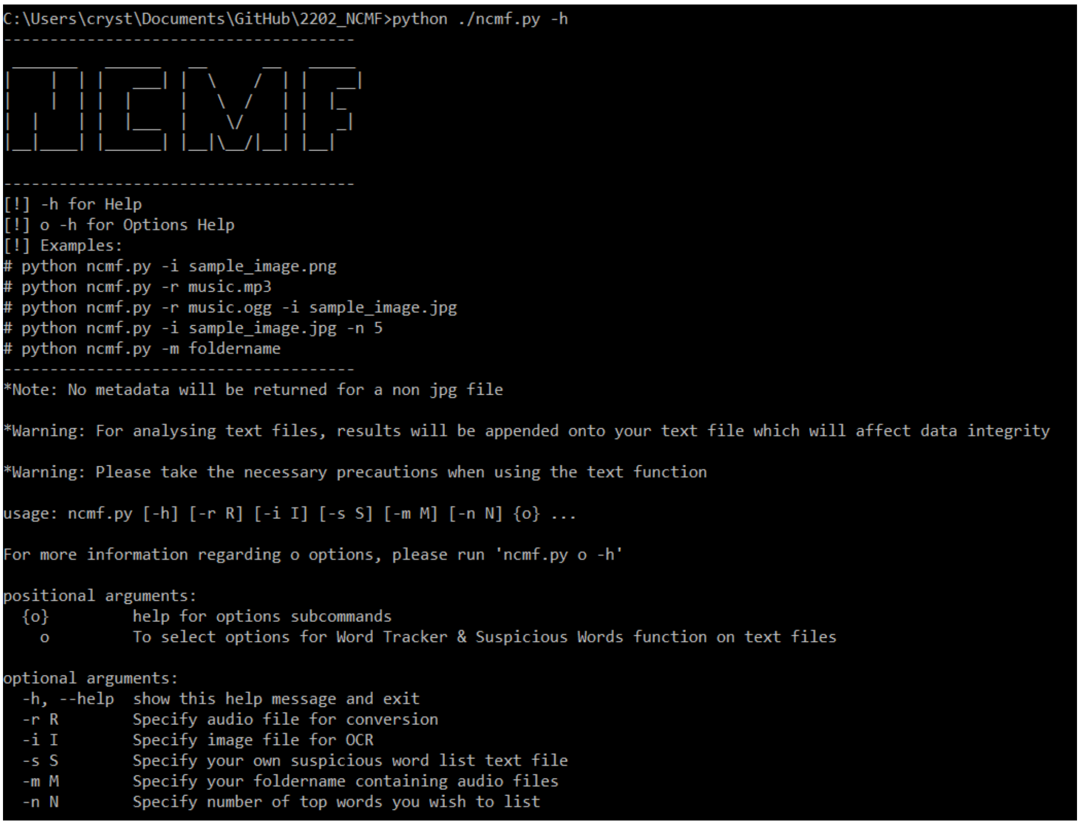
Fig 4

Main Help Page:

To see the available arguments in NCMF, enter the following command in your terminal:

- python ncmf.py -h

You should see the options appearing with the various functions you can perform with NCMF. Refer to table 1 for detailed explanation of what each function does.



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py -h
--------------------------------------
 _______  _______  __      __  ______
|   |   ||  _____||  \    /  ||  ____|
|   |   ||  |     |   \  /   ||  |_
|  |    ||  |____ |    \/    ||   _|
|__|____||_______||__|\__/|__||__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

usage: ncmf.py [-h] [-r R] [-i I] [-s S] [-m M] [-n N] {o} ...

For more information regarding o options, please run 'ncmf.py o -h'

positional arguments:
  {o}         help for options subcommands
    o         To select options for Word Tracker & Suspicious Words function on text files

optional arguments:
  -h, --help  show this help message and exit
  -r R        Specify audio file for conversion
  -i I        Specify image file for OCR
  -s S        Specify your own suspicious word list text file
  -m M        Specify your foldername containing audio files
  -n N        Specify number of top words you wish to list
```

Fig 5

Options	What it does
- h	Used to display the help menu
- r	Used when users want to analyze a single audio file
- i	Used when users wish to analyze an image file
- s	Used when users want to specify their own word list for the search against suspicious words with their audio transcription result or image text extraction results.
- m	Used when users want to analyze multiple audio files residing in a single folder
- n	Used when users want to modify the amount of top word occurrences to be displayed in the text results.

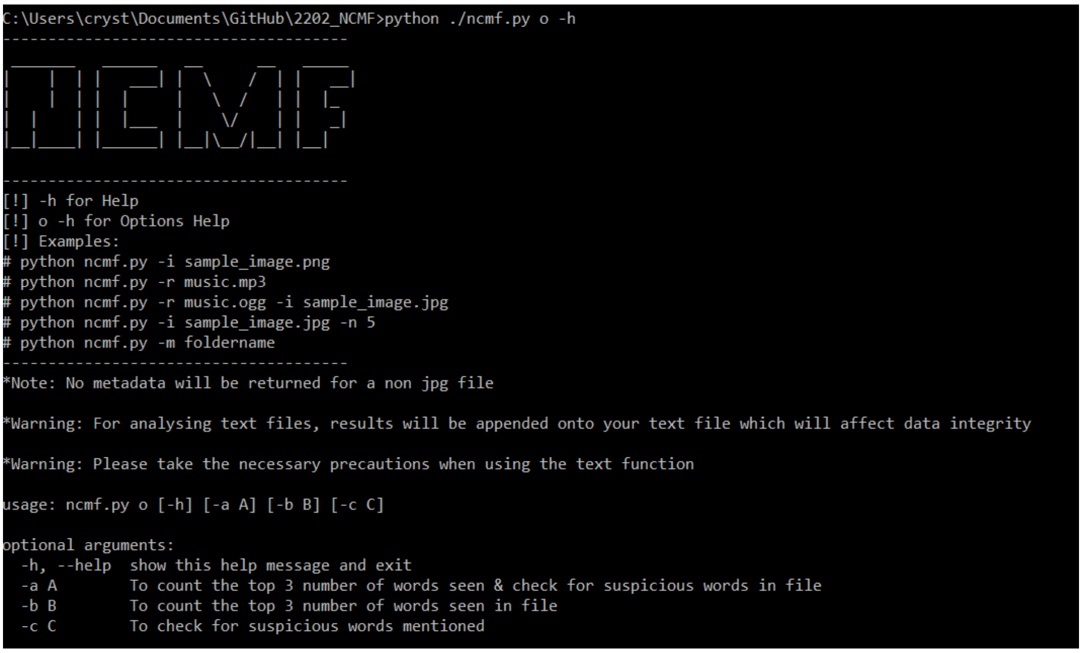
Table 1

Options for text-files:

NCMF supports analysis for text file as well in the event when users wish to check against a text file. To see the functions available for text file, type the following command in your terminal

- python ncmf.py o -h

Refer to Table 2 for detailed explanation of what each function does



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py o -h
--------------------------------------
 _______   ______  __      __   _____
|     | | |   ___||  \    /  ||  ___|
|     | | |  |    |   \  /   ||  |_
|  |    | |  |___ |    \/    ||   _|
|__|____| |______||__|\__/|__||__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

usage: ncmf.py o [-h] [-a A] [-b B] [-c C]

optional arguments:
  -h, --help  show this help message and exit
  -a A        To count the top 3 number of words seen & check for suspicious words in file
  -b B        To count the top 3 number of words seen in file
  -c C        To check for suspicious words mentioned
```

Fig 6

Options	What it does
o -a	Used when users want to analyze text files with the 2 functions (Counter and Sus_words)
o -b	Used when users only want to analyze text files to obtain the top 3 occurring words in the file
o -c	Used when users only want to analyze text files to obtain suspicious words discovered in the search against the tool's default suspicious word list

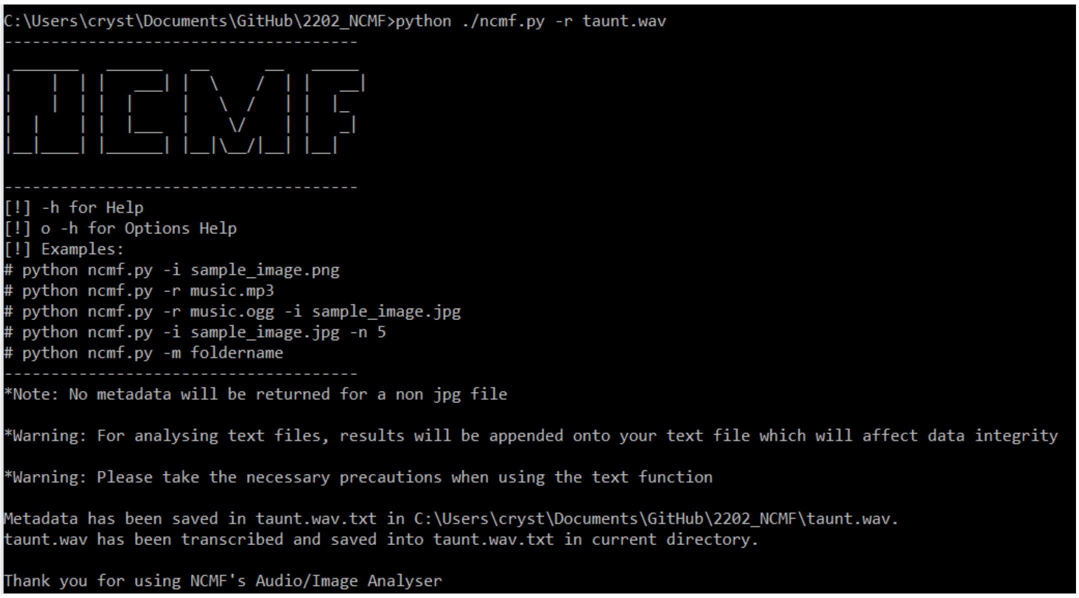
Table 2

4.1 Speech to Text Function

4.1.1 Providing only a single audio file

Step 1: Run the command 'python ncmf.py -r (filename)'

Step 2: Results are parsed in your current directory as printed by the tool



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py -r taunt.wav
--------------------------------------
 _______   ______  __     __  ______
|   |   | |  ____|| \   / || |  ___|
|   |   | | |     |  \ / | | | |_
|  |    | | |____ |   V  | | |  _|
|__|____| |______||__|\_/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

Metadata has been saved in taunt.wav.txt in C:\Users\cryst\Documents\GitHub\2202_NCMF\taunt.wav.
taunt.wav has been transcribed and saved into taunt.wav.txt in current directory.

Thank you for using NCMF's Audio/Image Analyser
```

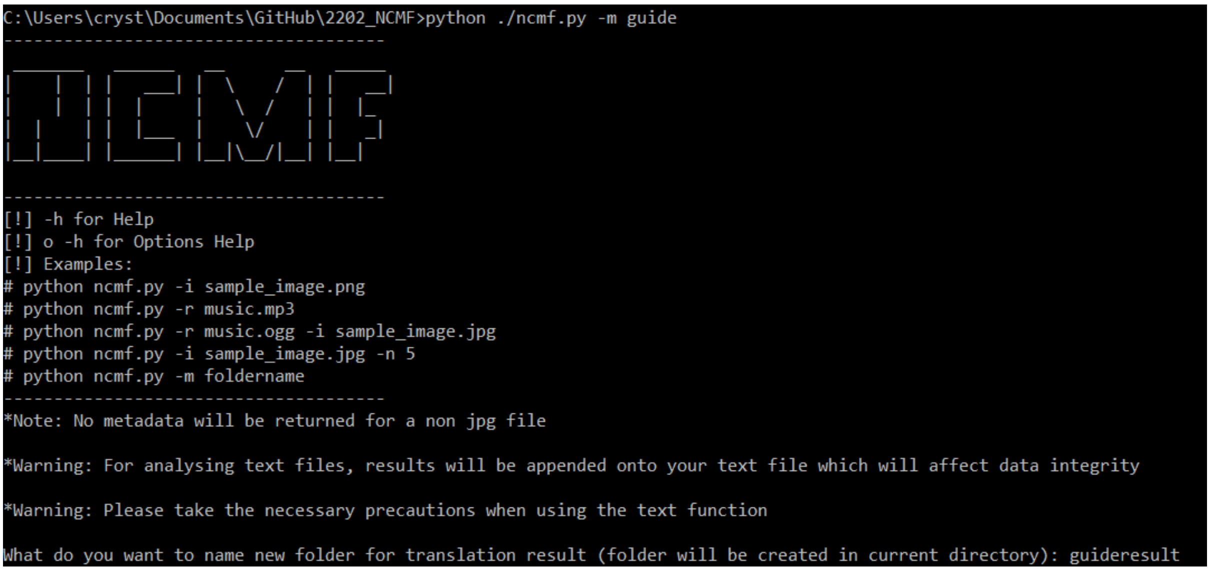
Fig 7. Sample command for Single Audio file

4.1.2 Providing a folder of multiple audio files

Step 1: Save all audio files into a single folder and save the folder into the directory with ncmf.py

Step 2: Run the command 'python ncmf.py -m (folder name)'

Step 3: You will be prompted to enter a name for a new folder. All generated results will be saved in the new folder created in the current directory.



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py -m guide
--------------------------------------
 _______   ______  __     __  ______
|   |   | |  ____|| \   / || |  ___|
|   |   | | |     |  \ / | | | |_
|  |    | | |____ |   V  | | |  _|
|__|____| |______||__|\_/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

What do you want to name new folder for translation result (folder will be created in current directory): guideresult
```

Fig 8. Sample output for Step 3

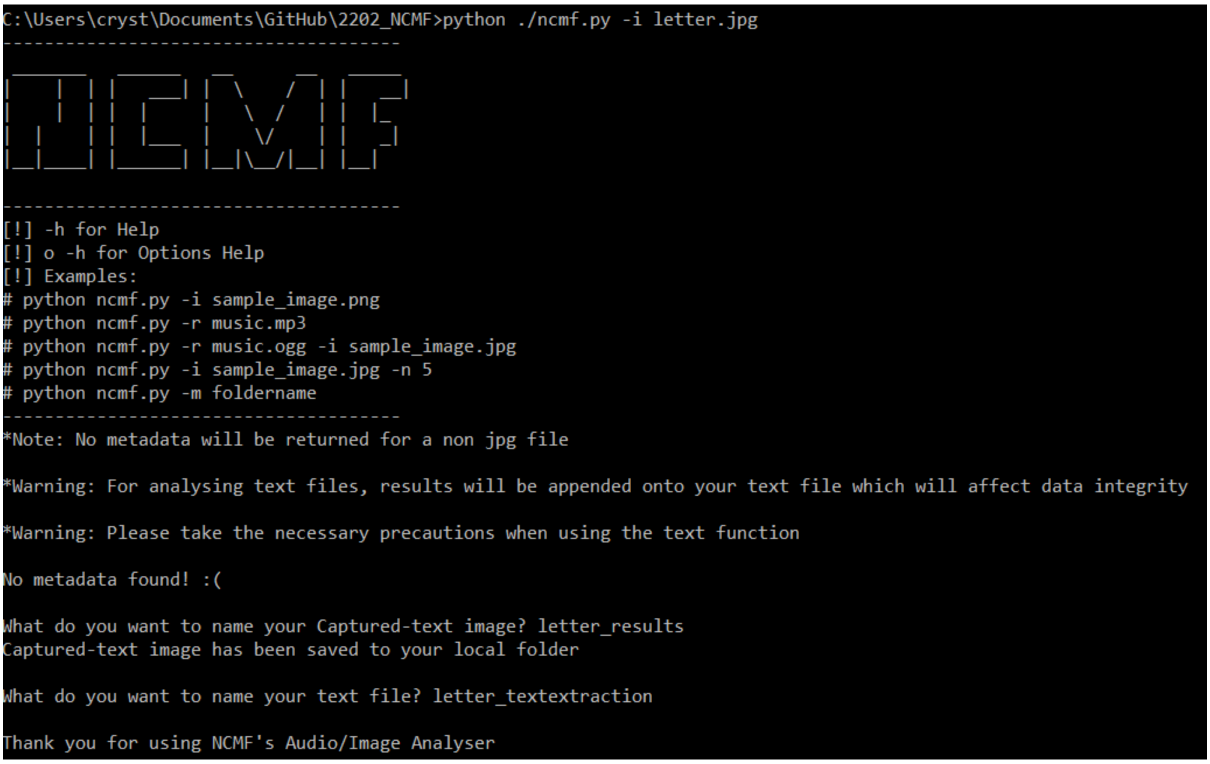
4.2 Image OCR Function

4.2.1 Text extraction on image files

Step 1: Run the command ‘python ncmf.py -i (image name)’

Step 2: You will be prompted to enter a name for 2 files which will be saved in the current directory:

1. Image consisting of captured text
2. Text file with extraction results



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py -i letter.jpg
--------------------------------------
 _______   ______   __      __   _____
|     | | |   ___| |  \    /  | |   __|
|     | | |  |     |   \  /   | |  |_
|  |    | |  |___  |    \/    | |   _|
|__|____| |______| |__|\__/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

No metadata found! :(

What do you want to name your Captured-text image? letter_results
Captured-text image has been saved to your local folder

What do you want to name your text file? letter_textextraction

Thank you for using NCMF's Audio/Image Analyser
```

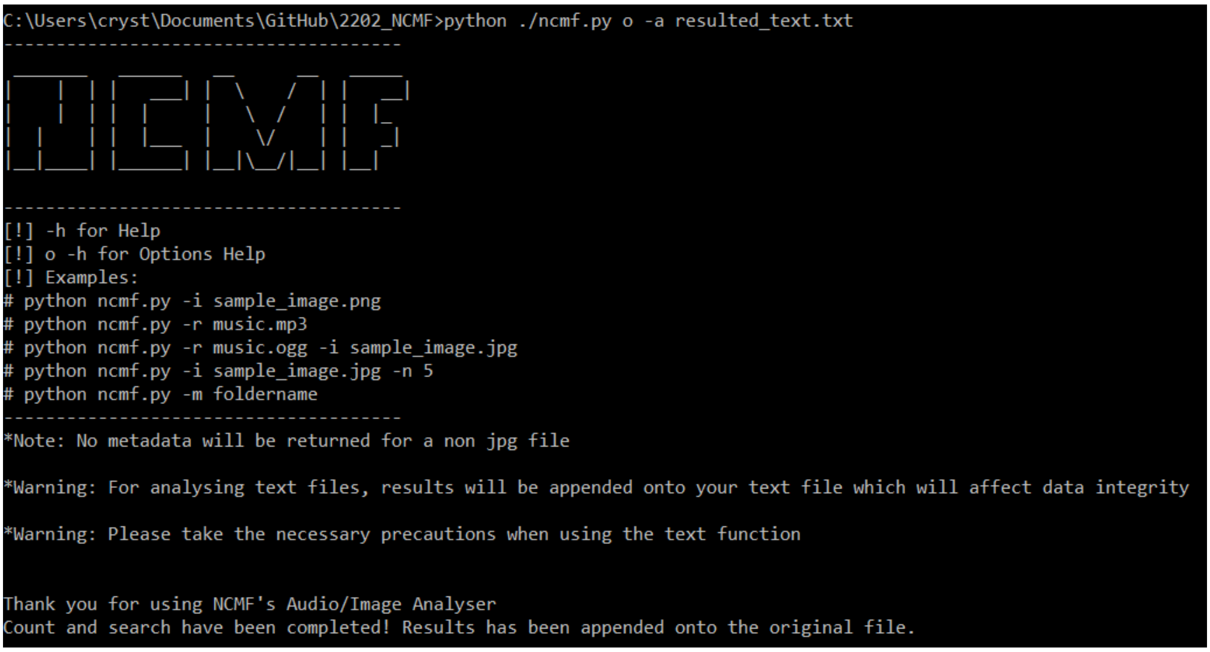
Fig 9. Sample output for step 2

4.3 Text file Function

4.3.1 Count the top 3 word occurrence & search for suspicious word list in a text file

Step 1: Run the command ‘python ncmf.py o -a (text filename)’

Step 2: When the tool has finished the count and search, the results will be appended to the provided text file.



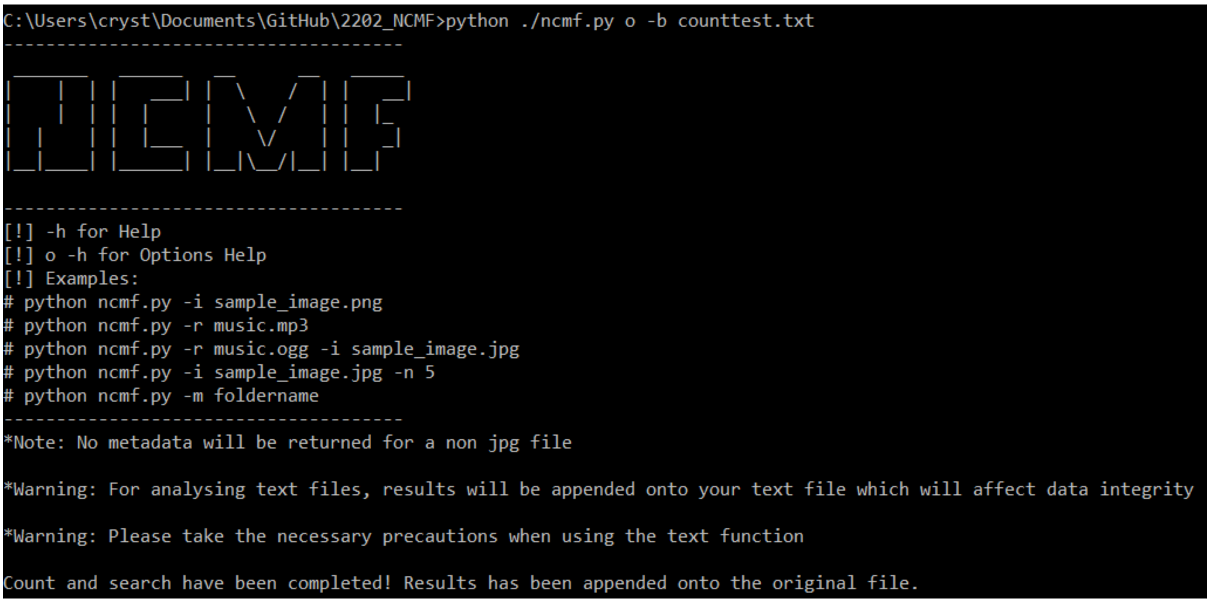
```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py o -a resulted_text.txt
--------------------------------------
 _______   ______   __    __   _____
|     | | |  ___| | \    / | | |  __|
|     | | | |     |  \  /  | | | |_
| |   | | | |___  |   \/   | | |  _|
|_|___| | |_____| |__|\__/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function


Thank you for using NCMF's Audio/Image Analyser
Count and search have been completed! Results has been appended onto the original file.
```

Fig 10. Sample output after tool has finished processing request

4.3.2 Count the top 3 word occurrence in a text file

Step 1: Run the command ‘python ncmf.py o -b (text filename)’

Step 2: When the tool has finished the count, the results will be appended to the provided text file.



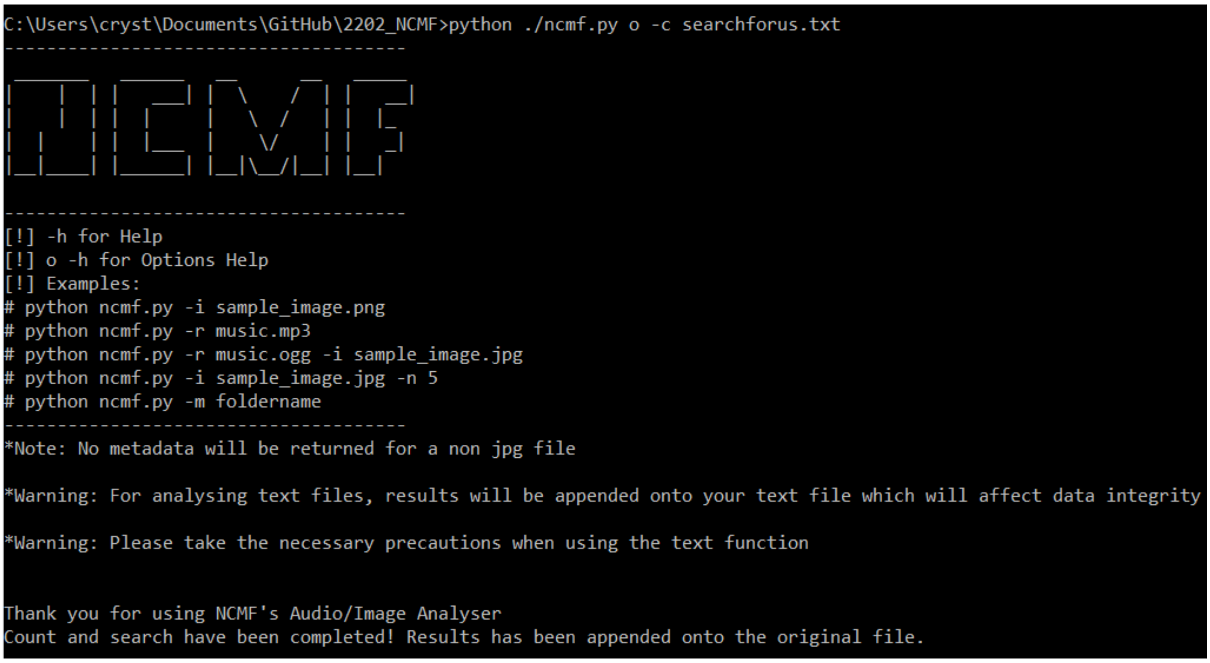
```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py o -b counttest.txt
--------------------------------------
 _______   ______   __    __   _____
|     | | |  ___| | \    / | | |  __|
|     | | | |     |  \  /  | | | |_
| |   | | | |___  |   \/   | | |  _|
|_|___| | |_____| |__|\__/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function

Count and search have been completed! Results has been appended onto the original file.
```

Fig 11. Sample output after tool has finished processing request

4.3.3 Search for suspicious word list in text files

Step 1: Run the command ‘python ncmf.py o -c (text filename)’

Step 2: When the tool has finished the search, the results will be appended to the provided text file.



```
C:\Users\cryst\Documents\GitHub\2202_NCMF>python ./ncmf.py o -c searchforus.txt
--------------------------------------
 _______   ______   __     __  _____
|     | | |  ___| | \    /  | |  __|
|     | | | |     |  \  /   | | |_
| |     | | |___  |   \/    | |  _|
|_|_____| |_____| |__|\__/|__| |__|

--------------------------------------
[!] -h for Help
[!] o -h for Options Help
[!] Examples:
# python ncmf.py -i sample_image.png
# python ncmf.py -r music.mp3
# python ncmf.py -r music.ogg -i sample_image.jpg
# python ncmf.py -i sample_image.jpg -n 5
# python ncmf.py -m foldername
--------------------------------------
*Note: No metadata will be returned for a non jpg file

*Warning: For analysing text files, results will be appended onto your text file which will affect data integrity

*Warning: Please take the necessary precautions when using the text function


Thank you for using NCMF's Audio/Image Analyser
Count and search have been completed! Results has been appended onto the original file.
```

Fig 12. Sample output after tool has finished processing request