Assignment: Audio Fingerprinting Task **Course**: MultiMedia Security(17685207)
Submitted to: Prof. Alberto Blanco Justicia
Team: Youssef Ezz, Kuldeep Parade Patil, Rahul Kumar, Hamza Abbas

Project Documentation

Modules and Dependencies

- numpy==1.24.0
- peewee==3.17.5
- scipy==1.13.1
- sqlite3
- os
- scipy.io
- scipy.ndimage
- hashlib
- time
- warnings
- argparse
- builddb
- IPython.display
- sounddevice
- matplotlib.pyplot
- Pandas
- sklearn.metrics

Installation

To install the required modules, run:

```
$       pip install -r requirements.txt --user
```

—Recommended to create and run in a python virtual environment:

```
$       python -m venv "path/to/desired/folder"

$       cd "path/to/desired/folder"

        //copy the files here

$       Scripts\activate

        // now you're in the virtual environment. Install requirements.txt
```

Design of Utilities

Programming Language
Python

- Reason: Python is an excellent choice for this project due to its extensive libraries for scientific computing, data manipulation, and audio processing, making it easier to implement and manage complex functionalities.

Parameters

- **the number of points in each segment of the signal**:
  - Samplesperseg(1024): Length of each segment
- **the number of points to overlap between segments**:
  - Samplesoverlap(1024 // 8) : same as default number of overlapping samples (samplesperseg // 8)
- **Window Size used to get constellation points**:
  - window_size(50) : the region centered around a time-frequency point such that it has a higher energy content than all its neighbors in that window or region
- **offset of the target zone**:
  - Targetzone_offset(0.1): offset in seconds to the right of an anchor point where the target zone begins
- **width of the target zone**:
  - Targetzone_offset(10): width of target zone in seconds
- **height of the target zone**:
  - Targetzone_offset(1000): height of target zone in frequency

Database Schemas

- Song Table:
  - id: Integer, Primary Key
  - title: Text
- Fingerprints Table:
  - id: Integer, Primary Key
  - song_id: Integer, Foreign Key (References Song Table)
  - fingerprint: Text //hash
  - anchortime: float //offset

Other Design Choices

- Details:
  -

User Manual

System Requirements

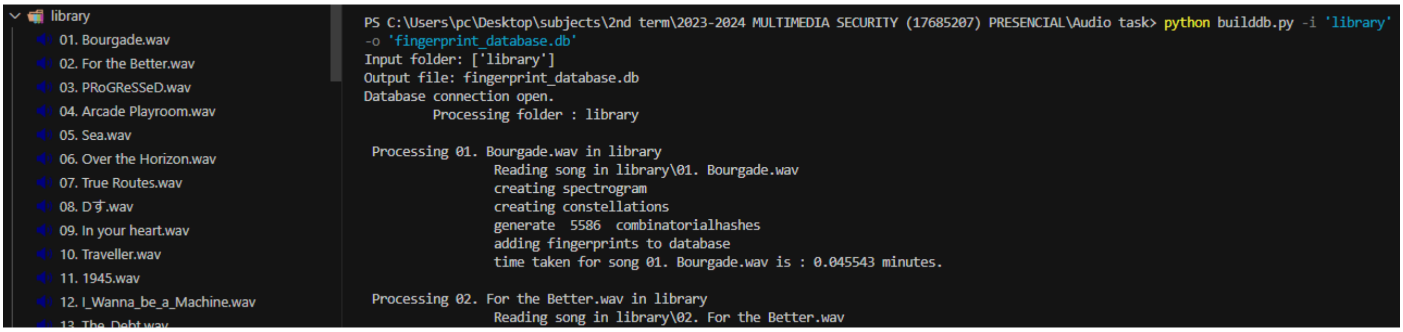
- Operating System: Windows, macOS, or Linux
- Python Version: 3.8 or higher
- Required Disk Space: Minimum 500MB

Running the Application

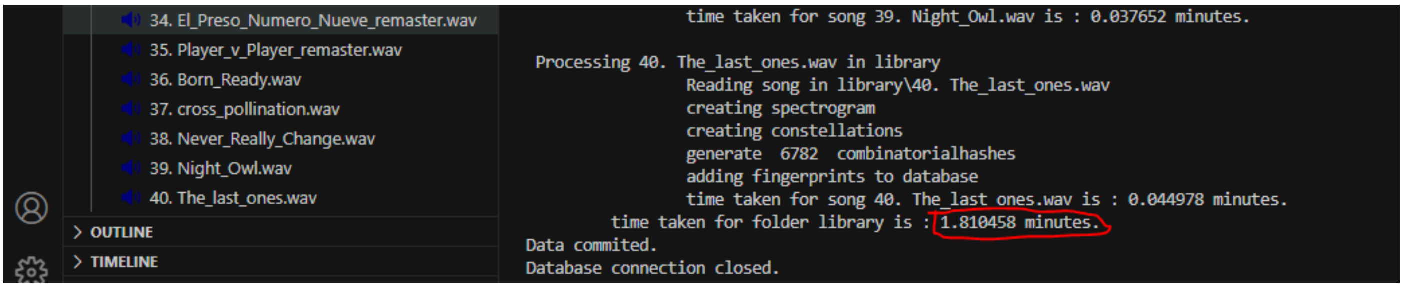
1. Install dependencies using the provided requirements.txt.

2. Run the database build script builddb.py t(takes about 1.8 minute):

$ python builddb.py -i 'library' -o 'fingerprint_database.db'



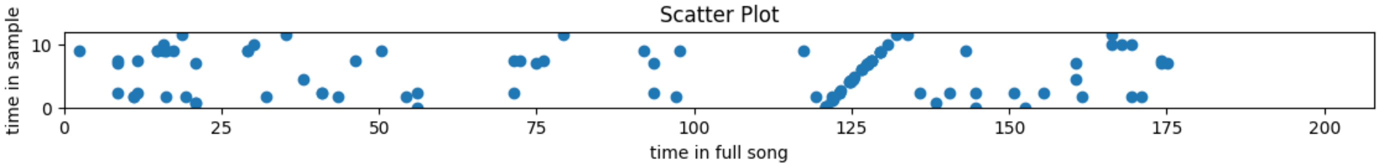
```
PS C:\Users\pc\Desktop\subjects\2nd term\2023-2024 MULTIMEDIA SECURITY (17685207) PRESENCIAL\Audio task> python builddb.py -i 'library'
-o 'fingerprint_database.db'
Input folder: ['library']
Output file: fingerprint_database.db
Database connection open.
         Processing folder : library

 Processing 01. Bourgade.wav in library
                 Reading song in library\01. Bourgade.wav
                 creating spectrogram
                 creating constellations
                 generate  5586  combinatorialhashes
                 adding fingerprints to database
                 time taken for song 01. Bourgade.wav is : 0.045543 minutes.

 Processing 02. For the Better.wav in library
                 Reading song in library\02. For the Better.wav
```

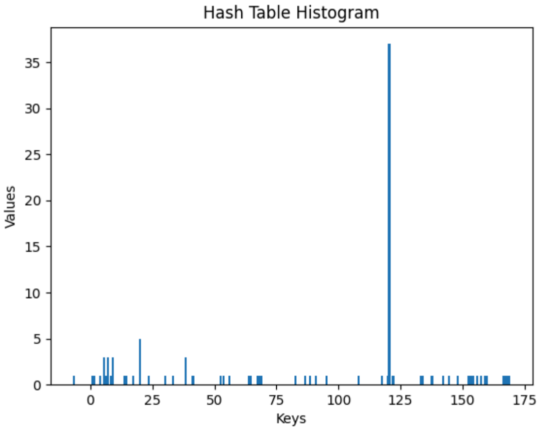

34. El_Preso_Numero_Nueve_remaster.wav
35. Player_v_Player_remaster.wav
36. Born_Ready.wav
37. cross_pollination.wav
38. Never_Really_Change.wav
39. Night_Owl.wav
40. The_last_ones.wav
OUTLINE
TIMELINE

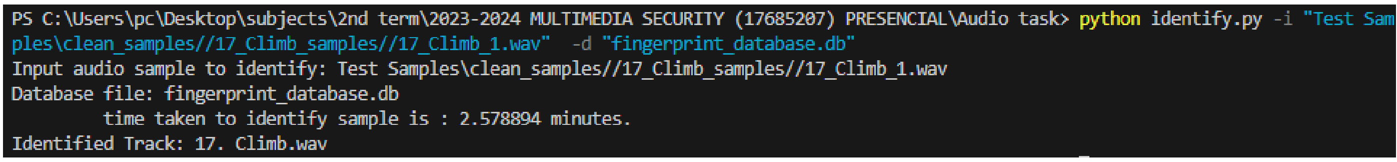
```
                 time taken for song 39. Night_Owl.wav is : 0.037652 minutes.

 Processing 40. The_last_ones.wav in library
                 Reading song in library\40. The_last_ones.wav
                 creating spectrogram
                 creating constellations
                 generate  6782  combinatorialhashes
                 adding fingerprints to database
                 time taken for song 40. The_last_ones.wav is : 0.044978 minutes.
         time taken for folder library is : 1.810458 minutes.
Data commited.
Database connection closed.
```

3. Run the identify sample script and optionally plot the scatter plot and histogram based on parameter **plot = True** in **identify.py**:

$ python identify.py -i "Test Samples\clean_samples//17_Climb_samples//17_Climb_1.wav" -d "fingerprint_database.db"



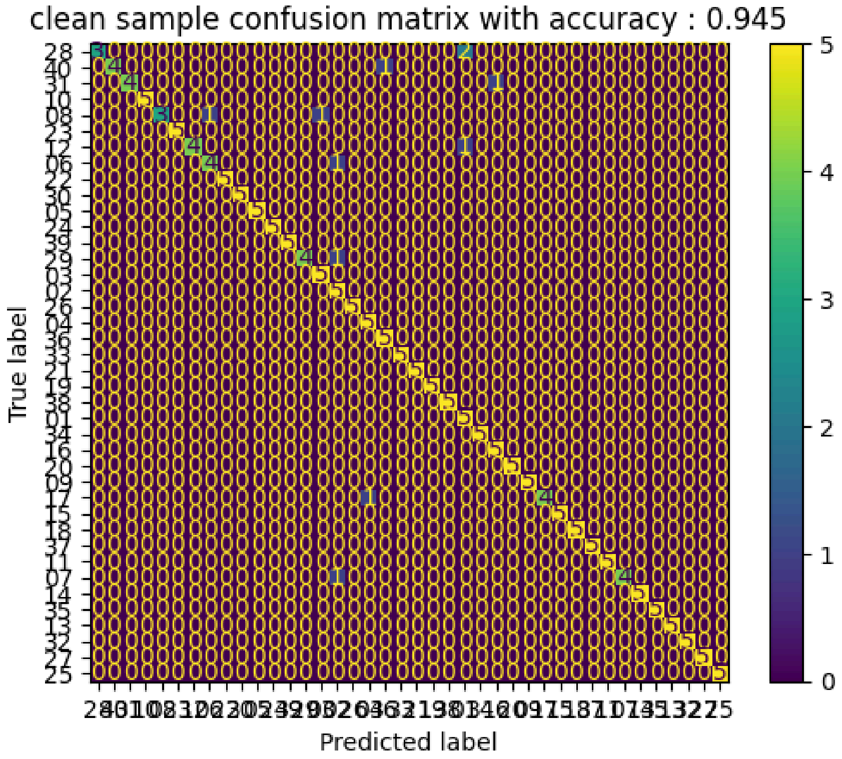




```
PS C:\Users\pc\Desktop\subjects\2nd term\2023-2024 MULTIMEDIA SECURITY (17685207) PRESENCIAL\Audio task> python identify.py -i "Test Sam
ples\clean_samples//17_Climb_samples//17_Climb_1.wav"  -d "fingerprint_database.db"
Input audio sample to identify: Test Samples\clean_samples//17_Climb_samples//17_Climb_1.wav
Database file: fingerprint_database.db
          time taken to identify sample is : 2.578894 minutes.
Identified Track: 17. Climb.wav
```

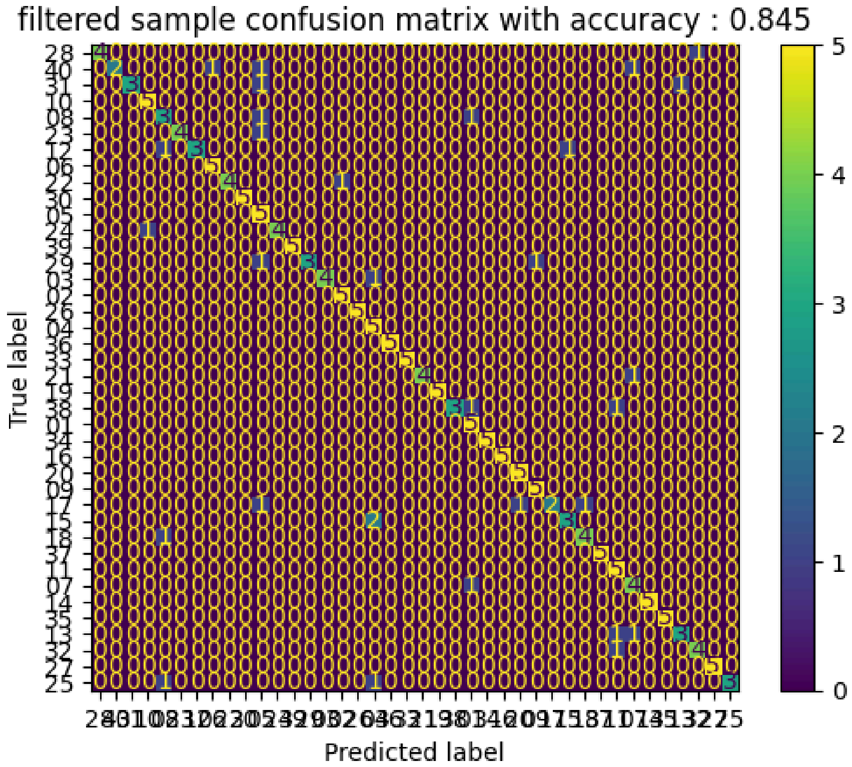
Detection Metrics

- Accuracy of classification `(TP + TN) / (TP + TN + FP + FN)`: see details in file **test_db.ipynb**
  - Details: classification is done for each category of the four
    - Clean sample

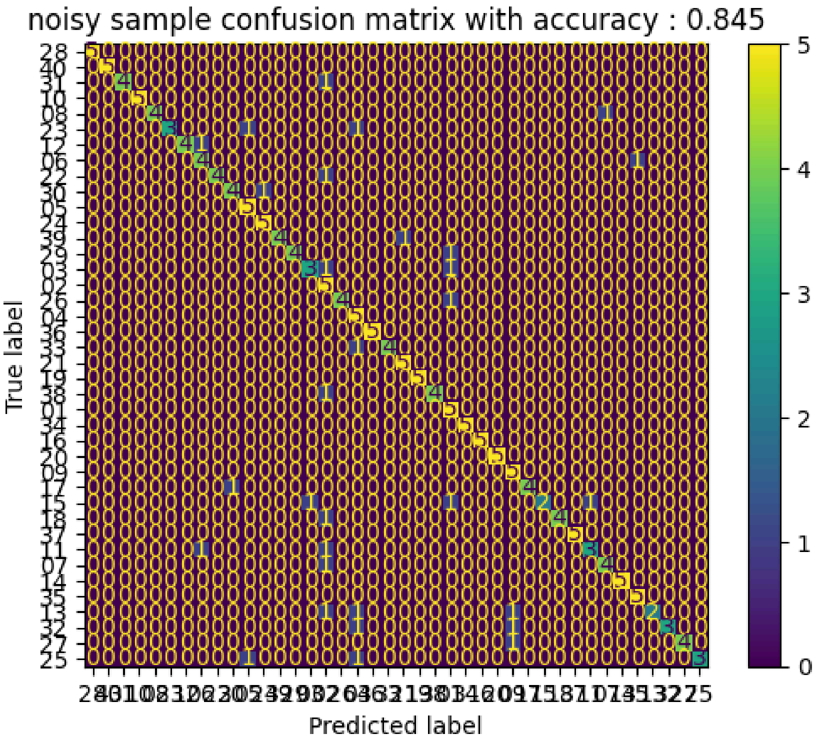


s

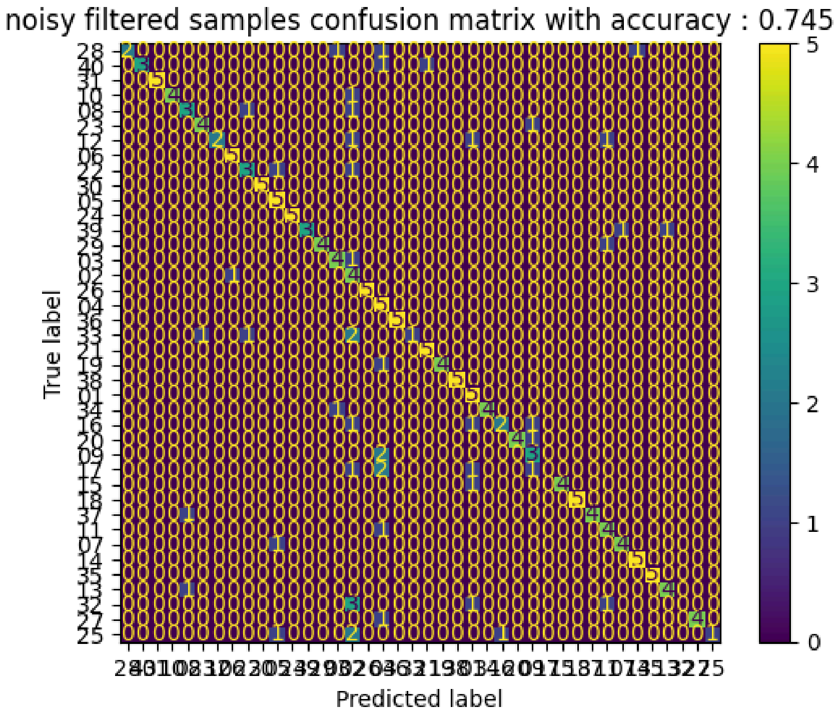
    - Filtered samples



- Noisy samples



- Noisy Filtered samples



- Time:
  - Details: total of 2 hours is the time taken for classification of all categories of samples (30 minutes each) see details in file **test_db.ipynb**

Features

- Identify songs using audio fingerprints.
- Store and manage song metadata.
- Generate and visualize song spectrograms.
- Classify different samples in different categories and calculate accuracy of classification

Troubleshooting

- Common issues and solutions:
  - Issue: Missing module error.
    - Solution: Ensure all dependencies are installed by running `pip install -r requirements.txt --user`.