A Project report

On

MUSIC INFORMATION RETRIEVAL SYSTEM USING OPENSOURCE TOOLS

Submitted By

Jagadeesh C	**(18BCS033)**
Arun Kumar S M	**(18BCS013)**
Anuj C S	**(18BCS009)**
Darshan R P	**(18BCS023)**

Under the guidance of

Dr. Uma Sheshadri

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DHARWAD

Table of contents

1. Introduction
    1.1. Aim and scope
    1.2. What is MIR?
    1.3. Why MIR?
2. About the project
    2.1. Things used in the project
    2.2. How MIR works (brief explanation)
    2.3. How MIR works
    2.4. Previously developed MIR applications
    2.5. Open-source algorithms
3. Review of literature
4. Report on the present investigation
5. Summary and conclusion
6. References
7. Acknowledgment

1. Introduction

While we listen to the excerpts, name as the many of its musical characteristics as we can. But, can you name the genre, tempo, instruments, mood or the time signature or the song structure?

The report consists of an overview on how clear a user gets to know about music Information retrieval system literature. The main intention was to make the information in the presentation become user friendly, maximum care has been taken by all four members of the team to make users know the information clearly and have a proper knowledge about music Information retrieval. There is a systematic explanation of each step and each topic with screenshots and python commands. We have been through a perfect flow of Abstract, Introduction, How MIR works and things used in the project, References and Acknowledgement and Conclusion.

▪ Aim and scope

The aim of this project is to develop a command line python programming language-based application, which is used to identify a music/song just by listening to a small portion of that song at any point.

▪ What is MIR

The main objective of this project is to help the user find the music/song or album just by listening to a small portion of the music.

The user mainly uses our product when they don’t know the name of the music they have been listening to, like the cases while watching a movie or a random video. Our application helps them to identify the music name just by listening to a portion of that music.

Here is a sampling of tasks that found in MIR:

- Cover song detection
- fingerprinting
- recommendation

    - genre recognition
    - instrument recognition
    - source separation
    - tempo estimation
    - pitch tracking
    - song structure/form
    - score alignment
    - query by humming
    - beat tracking
    - key detection

- **Why MIR?**
  - MIR used to discover and organize media collections.
  - It is used to search ("finding something that sounds like this") songs, speech, loops, sound effects and environmental sounds.
  - It is used to detect workflows in consumer products by the help of machine hearing.
  - It is used for the automatic control of mobile devices and software.

2. About the project

Things used in this project

- Python2.7 – is the programming language used in this project.
- Python-pip – python package manager used to add or remove packages.
- Python-tk – python library used to create GUI (graphical user interface).
- ffmpeg – it is a command line tool designed to process video and audio files.
- Portaudio – is a library used for recording and audio playbacks.
- Pyaudio - provides Python bindings for PortAudio, With PyAudio, you can easily use Python to record audio and play on a variety of platforms.
- Matplotlib – is a python library used for data visualization and graphical plotting library for Python and its numerical extension NumPy.
- Termcolor – it is a python module used for Color formatting for output text in the terminal.
- Scipy - Python library used for technical computing and scientific computing.
- Pydub – it is a python library used to split, play, edit and merge audio files.

How MIRS works (brief explanation)

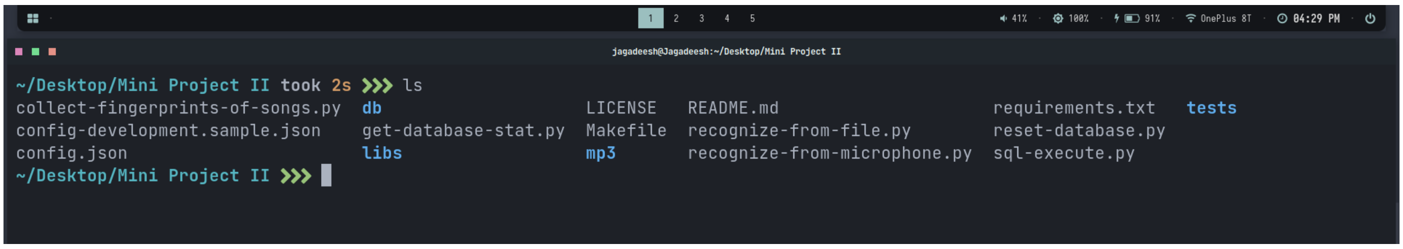
MIRS identifies songs based on an **“audio fingerprint”.**

- MIRS already has the audio fingerprint of the song/music stored in the database.

  (audio fingerprint – is like a unique id for each song to be able to recognize it, which is created using an audio files frequency with respect to time and other criteria for the whole song, music or audio)
- For every new song, album, music released MIRS requires to create audio fingerprint of those songs and store it in the database. (It’s the work of the product owner or the people maintaining the MIRS to create fingerprint of the newly released song, album and music in order for the user to find them.)
- It tags a song for 5-10 seconds and the application creates an audio fingerprint for a particular song.
- For any user wants to use the application, it’ll uses their computer or smartphone’s built-in microphone to gather/collect a brief sample of audio being played.
- MIRS works by analyzing the captured sound and seeking/finding a match based on an audio fingerprint in the database of millions of songs.
- If it finds a match, it sends/returns information such as the artist, song title, and album back to the user.
- For our application it’ll send/return the song name.

▪ How MIRS works

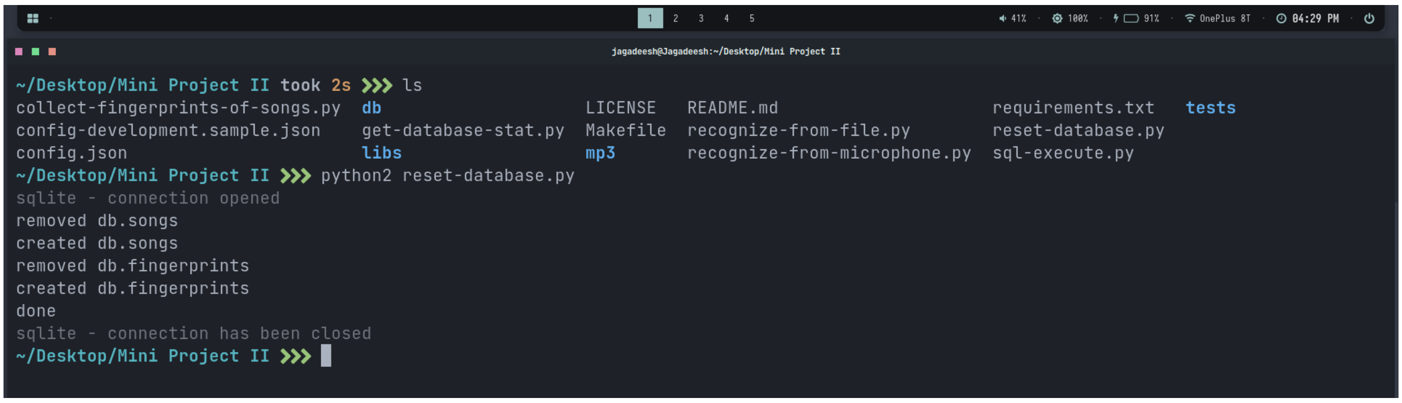
Music information retrieval system is a **command line python application** which returns the name of the song just by listening to a small portion of that song.



```
jagadeesh@Jagadeesh:~/Desktop/Mini Project II
~/Desktop/Mini Project II took 2s ❯❯❯ ls
collect-fingerprints-of-songs.py  db                    LICENSE   README.md                  requirements.txt  tests
config-development.sample.json    get-database-stat.py  Makefile  recognize-from-file.py     reset-database.py
config.json                       libs                  mp3       recognize-from-microphone.py  sql-execute.py
~/Desktop/Mini Project II ❯❯❯
```

▪ Python2 reset-database.py

“**reset-database.py**” is a python script used to reset the database containing audio fingerprints. Generally used to clear the previous songs audio fingerprints.

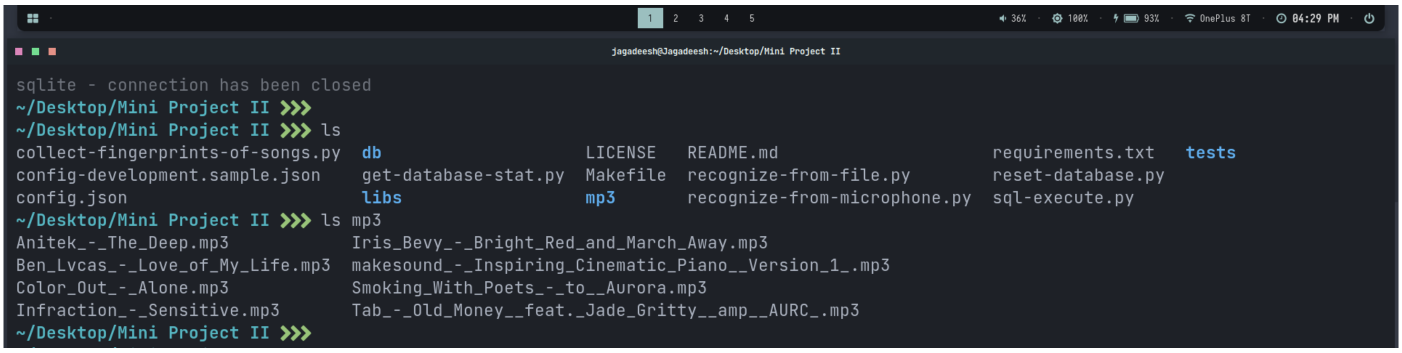


```
jagadeesh@Jagadeesh:~/Desktop/Mini Project II
~/Desktop/Mini Project II took 2s ❯❯❯ ls
collect-fingerprints-of-songs.py  db                    LICENSE   README.md                  requirements.txt  tests
config-development.sample.json    get-database-stat.py  Makefile  recognize-from-file.py     reset-database.py
config.json                       libs                  mp3       recognize-from-microphone.py  sql-execute.py
~/Desktop/Mini Project II ❯❯❯ python2 reset-database.py
sqlite - connection opened
removed db.songs
created db.songs
removed db.fingerprints
created db.fingerprints
done
sqlite - connection has been closed
~/Desktop/Mini Project II ❯❯❯
```

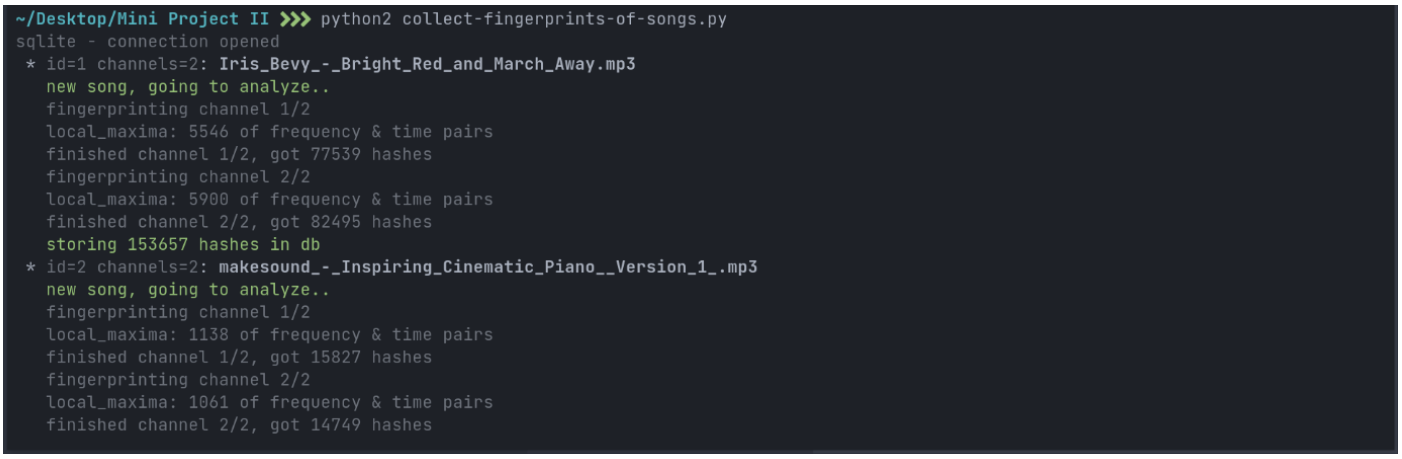
- **python2 collect-fingerprints-of-songs.py**

"collect-fingerprits-of-songs.py" is a python script used to create fingerprints of **.mp3** of songs stored in **mp3** folder.

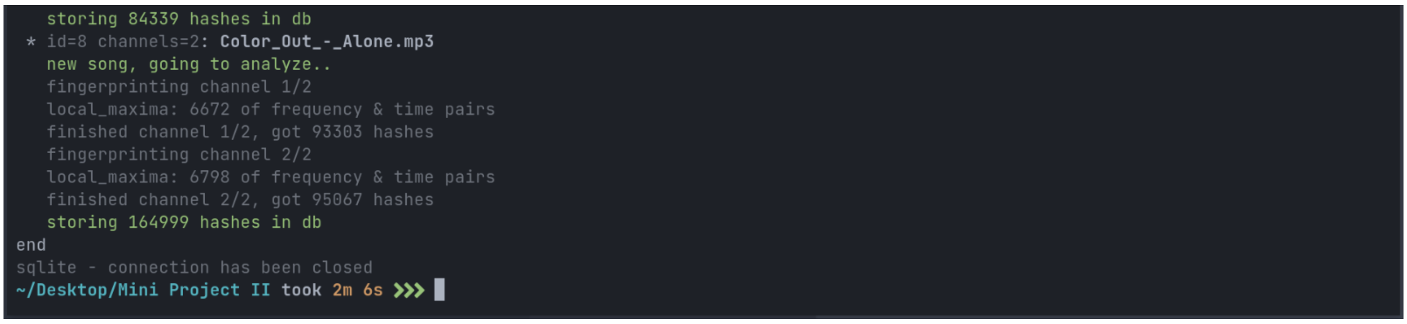
This application will look for **.mp3** songs stored in the folder named **mp3** which is located in the same directory as our application and creates fingerprint of those songs and store them in the database, so that if anyone using our application would be able to find the song just by playing a small portion of those songs.



```
sqlite - connection has been closed
~/Desktop/Mini Project II ❯❯❯
~/Desktop/Mini Project II ❯❯❯ ls
collect-fingerprints-of-songs.py  db                    LICENSE   README.md                    requirements.txt  tests
config-development.sample.json    get-database-stat.py  Makefile  recognize-from-file.py       reset-database.py
config.json                       libs                  mp3       recognize-from-microphone.py sql-execute.py
~/Desktop/Mini Project II ❯❯❯ ls mp3
Anitek_-_The_Deep.mp3            Iris_Bevy_-_Bright_Red_and_March_Away.mp3
Ben_Lvcas_-_Love_of_My_Life.mp3  makesound_-_Inspiring_Cinematic_Piano__Version_1_.mp3
Color_Out_-_Alone.mp3            Smoking_With_Poets_-_to__Aurora.mp3
Infraction_-_Sensitive.mp3       Tab_-_Old_Money__feat._Jade_Gritty__amp__AURC_.mp3
~/Desktop/Mini Project II ❯❯❯
```



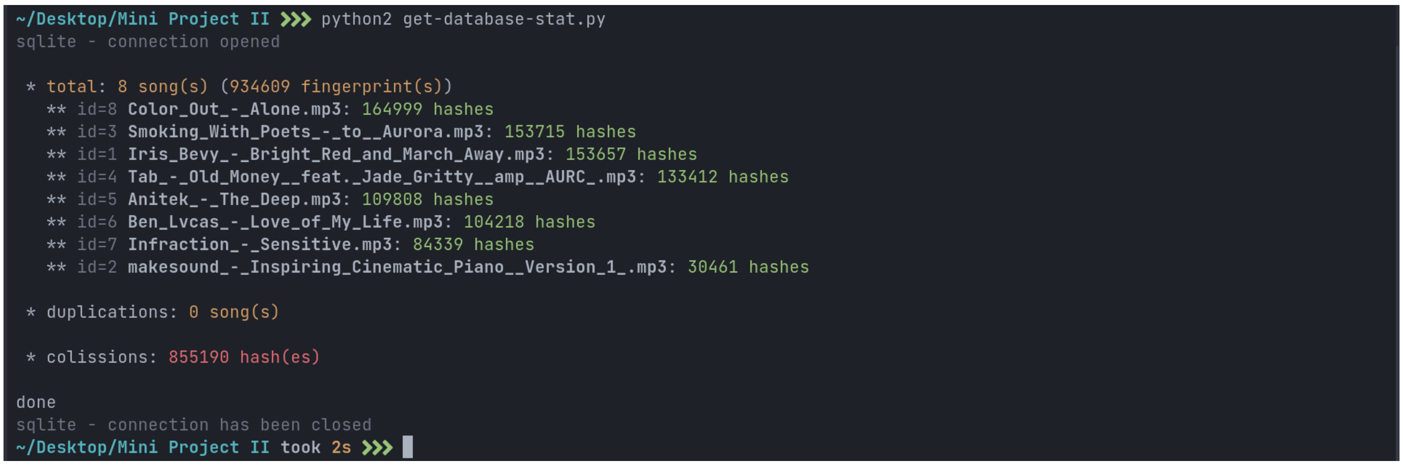
```
~/Desktop/Mini Project II ❯❯❯ python2 collect-fingerprints-of-songs.py
sqlite - connection opened
 * id=1 channels=2: Iris_Bevy_-_Bright_Red_and_March_Away.mp3
   new song, going to analyze..
   fingerprinting channel 1/2
   local_maxima: 5546 of frequency & time pairs
   finished channel 1/2, got 77539 hashes
   fingerprinting channel 2/2
   local_maxima: 5900 of frequency & time pairs
   finished channel 2/2, got 82495 hashes
   storing 153657 hashes in db
 * id=2 channels=2: makesound_-_Inspiring_Cinematic_Piano__Version_1_.mp3
   new song, going to analyze..
   fingerprinting channel 1/2
   local_maxima: 1138 of frequency & time pairs
   finished channel 1/2, got 15827 hashes
   fingerprinting channel 2/2
   local_maxima: 1061 of frequency & time pairs
   finished channel 2/2, got 14749 hashes
```



```
   storing 84339 hashes in db
 * id=8 channels=2: Color_Out_-_Alone.mp3
   new song, going to analyze..
   fingerprinting channel 1/2
   local_maxima: 6672 of frequency & time pairs
   finished channel 1/2, got 93303 hashes
   fingerprinting channel 2/2
   local_maxima: 6798 of frequency & time pairs
   finished channel 2/2, got 95067 hashes
   storing 164999 hashes in db
end
sqlite - connection has been closed
~/Desktop/Mini Project II took 2m 6s ❯❯❯
```

- **python2 get-database-stat.py**

  “get-database-stat.py” is a python script used to get the list of names of the songs whose fingerprint has been created and stored in the database using **“collect-fingerprints-of-songs.py”** python script.



```
~/Desktop/Mini Project II »» python2 get-database-stat.py
sqlite - connection opened

 * total: 8 song(s) (934609 fingerprint(s))
   ** id=8 Color_Out_-_Alone.mp3: 164999 hashes
   ** id=3 Smoking_With_Poets_-_to__Aurora.mp3: 153715 hashes
   ** id=1 Iris_Bevy_-_Bright_Red_and_March_Away.mp3: 153657 hashes
   ** id=4 Tab_-_Old_Money__feat._Jade_Gritty__amp__AURC_.mp3: 133412 hashes
   ** id=5 Anitek_-_The_Deep.mp3: 109808 hashes
   ** id=6 Ben_Lvcas_-_Love_of_My_Life.mp3: 104218 hashes
   ** id=7 Infraction_-_Sensitive.mp3: 84339 hashes
   ** id=2 makesound_-_Inspiring_Cinematic_Piano__Version_1_.mp3: 30461 hashes

 * duplications: 0 song(s)

 * colissions: 855190 hash(es)

done
sqlite - connection has been closed
~/Desktop/Mini Project II took 2s »»
```

- **python2 recognize-from-microphone.py –s n**

  “recognize-from-microphone.py” is a python script the user use to find the music.

  “recognize-from-microphone.py” will use user’s mobile phone or computer’s internal microphone to gather brief sample of audio as input and returns the name of the song being played.

Options:

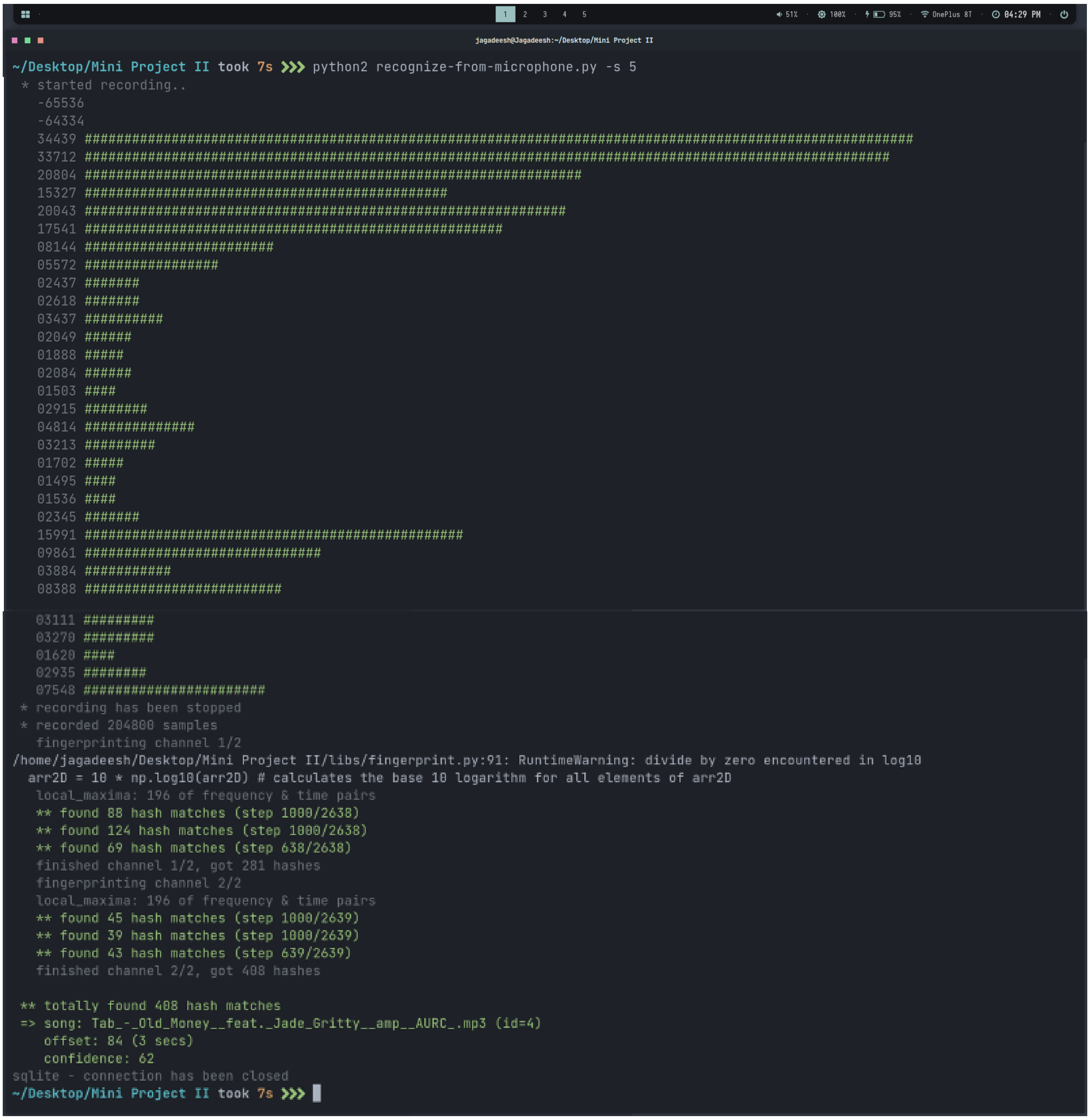
-s - defines how many seconds we want the script to listen

n – takes an integer value, defines how many seconds the application will listen for an audio.

Example:

python2 recognize-from-microphone.py -s 10

will listen for 10 seconds



```
jagadeesh@Jagadeesh:~/Desktop/Mini Project II

~/Desktop/Mini Project II took 7s ❯❯❯ python2 recognize-from-microphone.py -s 5
  * started recording..
    -65536
    -64334
    34439 #########################################################################################################
    33712 ######################################################################################################
    20804 ###############################################################
    15327 ##############################################
    20043 #############################################################
    17541 #####################################################
    08144 ########################
    05572 #################
    02437 #######
    02618 #######
    03437 ##########
    02049 ######
    01888 #####
    02084 ######
    01503 ####
    02915 ########
    04814 ##############
    03213 #########
    01702 #####
    01495 ####
    01536 ####
    02345 #######
    15991 ################################################
    09861 ##############################
    03884 ###########
    08388 #########################
    03111 #########
    03270 #########
    01620 ####
    02935 ########
    07548 #######################
  * recording has been stopped
  * recorded 204800 samples
    fingerprinting channel 1/2
/home/jagadeesh/Desktop/Mini Project II/libs/fingerprint.py:91: RuntimeWarning: divide by zero encountered in log10
  arr2D = 10 * np.log10(arr2D) # calculates the base 10 logarithm for all elements of arr2D
    local_maxima: 196 of frequency & time pairs
    ** found 88 hash matches (step 1000/2638)
    ** found 124 hash matches (step 1000/2638)
    ** found 69 hash matches (step 638/2638)
    finished channel 1/2, got 281 hashes
    fingerprinting channel 2/2
    local_maxima: 196 of frequency & time pairs
    ** found 45 hash matches (step 1000/2639)
    ** found 39 hash matches (step 1000/2639)
    ** found 43 hash matches (step 639/2639)
    finished channel 2/2, got 408 hashes

  ** totally found 408 hash matches
  => song: Tab_-_Old_Money__feat._Jade_Gritty__amp__AURC_.mp3 (id=4)
     offset: 84 (3 secs)
     confidence: 62
sqlite - connection has been closed
~/Desktop/Mini Project II took 7s ❯❯❯ ▌
```

for the song **Tab_old_money__feat.._Jade_Gritty__amp__AURC.mp3** being played it returned the name of that song.

Note:

Any user only uses this application when he/she doesn’t know the name of a particular song/music, and our application helps them to find it.

Just for the demonstration,

We mentioned the name of the song in the beginning itself, so that we could verify the application detects the song correctly.

▪ Previously developed MIR applications

MIR is a small but growing field of research with many real-world applications.

There are several apps/applications developed on the idea of MIRS. like,

- **Shazam**

  Shazam is an application that can identify music, advertising, movies and television shows, based on a short sample audio played and using the microphone on the device.

  It was created by London-based Shazam Entertainment, and has been owned by Apple Inc. since 2018.

  Link: https://play.google.com/store/apps/details?id=com.shazam.android&hl=en_IN&gl=US

- **SoundHound**

  SoundHound Inc. is an audio and speech recognition company founded in 2005. It develops speech recognition, natural language understanding, sound recognition and search technologies.

  Link: https://play.google.com/store/apps/details?id=com.melodis.midomiMusicIdentifier.freemium&hl=en_IN&gl=US

- **Audiggle**
- **Audio Tag**

And many more.

Open-source algorithms

There are more than 198 open source algorithms are developed for MIRS. Like,

- **Essentia**

  C++ library for audio and music analysis, description and synthesis, including Python bindings.

  http://essentia.upf.edu/

- **Musicinformationretrieval.com**

  Instructional notebooks on music information retrieval.

  http://musicinformationretrieval.com/

- **Meyda**

  Audio feature extraction for JavaScript.

  https://github.com/meyda/meyda

- **Madmom**

  Python audio and music signal processing library

  https://madmom.readthedocs.io/

- **Strugatzki**

  Algorithms for matching audio file similarities https://github.com/Sciss/Strugatzki

- **Audio-fingerprint-python project**

  Fingerprint audio files & identify what's playing

https://github.com/itspoma/audio-fingerprint-identifying-python

Links for more – https://awesomeopensource.com/projects/music-information-retrieval

For our project we are using the **"Audio-fingerprint-python"** project https://github.com/itspoma/audio-fingerprint-identifying-python

This project is **shazam-similar** app, that identifies song using **audio fingerprinting.**

3. Review of literature

- Music Information Retrieval: Recent Developments and Applications

Music Information Retrieval: Recent Developments and applications is a literature written by

Markus Schedl (Johannes Kepler University Linz, Austria)
Emilia Gómez (Universitat Pompeu Fabra, Barcelona, Spain)
Julián Urbano (Universitat Pompeu Fabra, Barcelona, Spain)

Which give a check/survey of the field of MIRS, paying attention to the developments made lately, similar as the auto-tagging and the semantic and the user centric retrieval approaches.

First written on well-established and the proven styles for feature extraction and the music indexing, from both the contextual data sources and the audio signals about the music, similar as cooperative/collaborative tags or the web sites. These in turn enable the wide kinds of music retrieval tasks, similar as music identification ("query by illustration") or semantic music hunt/search. And they reviewed current work on modeling in the environment/context of music recommendation and user analysis and retrieval, addressing the recent trend towards user-centric and adaptive approaches and systems. And it follows the discussion about how different/various Music Information retrieval approaches to different problems are compared and evaluated/estimated.

4. Report on the present investigation

The status of AR systems is covered in the Survey of Music Information Retrieval systems, presented at the Sixth International Conference on Music Information Retrieval. In illustrating a summary of ‘Music Information Retrieval (MIR)’, a distinction is made between the content-based search systems of general ‘audio data’ and search systems for ‘music based on the notes.’ Alongside these are the ‘hybrid’ systems, which in the early treatment of any type of audio data were converted into a symbolic version of the notes.

With reference to music databases, content-based search has different perspectives. Search-by-humming allows users to search for pieces by humming, or strumming from memory. The traditional search-by-example, according to the type of similarity required, is useful for musicologists searching for pieces inspired by a melody. Lastly come searches orientated towards comparing whole soundtracks or their parts, proving useful in ‘investigations’ for copyright purposes into cases of plagiarism or quotation. AR techniques have numerous practical applications: identifying songs transmitted by broadcasters, also via a ‘common receiver’ connected to a treatment system; search for ‘suspicious’ sounds recorded by surveillance systems; and sound analysis of video and any type of application in television, radio or other media industry archive. Despite the novelty of its application, AR is making tasks faster and more efficient, and its applications are now present in a lot of commercial equipment.

5. Summary and Conclusions

There has been a lot of discussions on music Information retrieval system till date but not that much researches have been done on the same. There are only a few numbers of algorithms which are developed for music Information retrieval system. We got an idea of what kind of music being listened by what kind of age groups in a particular situation or environment through these online resources or pdfs, and have studied the algorithms that have been given in the article.

We got to know about the working of a MIR on detecting a song and providing information on it. There has been a systematic explanation about each and every steps involved in detection of a song, we studied the fingerprint matching algorithm and have used the same algorithm in our project.

Music Information Retrieval is a young but the established multidisciplinary field of research. Even the though the origin of the Music Information Retrieval system (MIR) can be tracked back to the 1960, the first International Conference on the Music Information Retrieval system, started in 2000 and has exerted on the sense of belongingness to the research
community.

Although the field is constantly evolving, there already exists a set of techniques that have become the standard in certain applications. In this report, we provided an introduction to MIRS and detailed overview of how to use our provided application.

A great variety of different methods for searching in the audio data and music scores has been proposed and implemented in the research prototypes and commercial systems. Besides the limited and well-defined tasks of the identifying recordings, for which the audio fingerprinting techniques works well, it is hard to tell that which methods should be further pursued.

User studies have been identified as the key components of music information research.

While the existing work has been provided the valuable findings and the recommendations for future MIRS development, expanded research attention will be required to provide a comprehensive picture of music information use.

6. References

https://musicinformationretrieval.com/index.html

https://github.com/itspoma/audio-fingerprint-identifying-python

https://aclanthology.org/L16-1312.pdf

https://www.researchgate.net/publication/220722985_User_studies_in_the_Music_Information_Retrieval_Literature

https://link.springer.com/article/10.1007%252Fs10844-013-0247-6

https://www.sciencedirect.com/topics/computer-science/music-information-retrieval

https://julian-urbano.info/files/publications/059-music-information-retrieval-recent-developments-applications.pdf

7. Acknowledgment

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.