

A Project report submitted in partial fulfilment of the requirement for the degree of
Bachelor of Technology in Computer Science Engineering
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. Overview
2. Thing used in this project
3. Working
 1. Brief explanation
 2. Detailed overview
4. Previously developed MIRS
5. Open source algorithms
6. References

OVERVIEW

The main objective of this project is, it helps the user to identify name of the music/song or album just by listening to a small portion of that music.

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 audio and video files.
- Portaudio – is a library used for audio playbacks and recording.
- Pyaudio - provides Python bindings for PortAudio, With PyAudio, you can easily use Python to play and record audio on a variety of platforms.

THINGS USED IN THIS PROJECT

- Matplotlib – is a python library used to 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 in the terminal.
- Scipy - Python library used for scientific computing and technical computing.
- Pydub – it is a python library used to play, split, merge, and edit audio files.

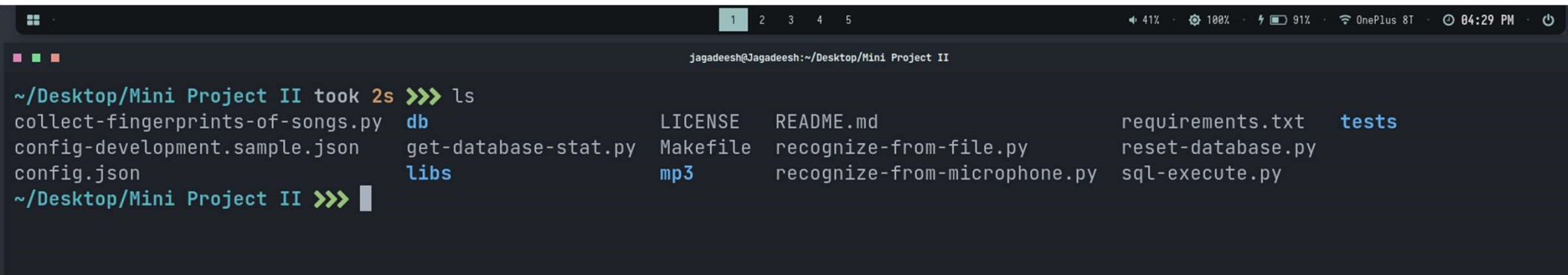
HOW MIRS WORKS? (BRIEF EXPLANATION).

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

- It uses a smartphone or computer's built-in microphone to gather a brief sample of audio being played.
- MIRS stores a catalogue of audio fingerprints in a database.
- The user tags a song for 10 seconds and the application creates an audio fingerprint.
- MIRS works by analyzing the captured sound and seeking a match based on an acoustic fingerprint in a database of millions of songs.
- If it finds a match, it sends information such as the artist, song title, and album back to the user.

HOW IT 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 >>> 
```

HOW IT WORKS

➤ Python2 reset-database.py

Use to reset the database

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

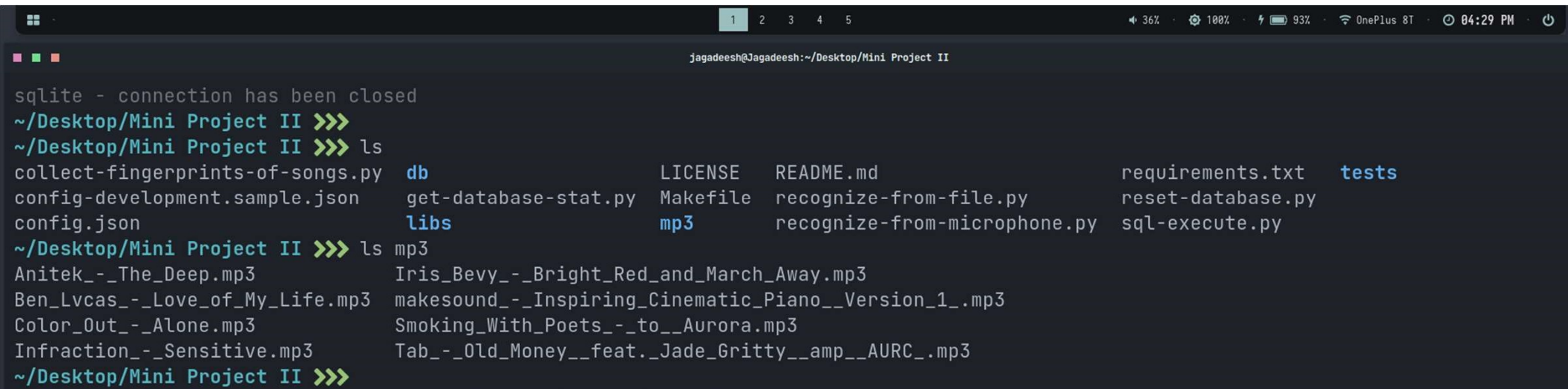
~/Desktop/Mini Project II took 2s >>> ls
collect-fingerprints-of-songs.py  db          LICENSE    README.md      requirements.txt  tests
config-development.sample.json    get-database-stat.py  Makefile    recognize-from-file.py  reset-database.py
config.json                      libs        mp3     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


HOW IT WORKS

> python2 collect-fingerprints-of-songs.py

it is a python script used to **create fingerprints** of **.mp3** songs stored in **mp3** folder



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

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
Ben_Lvcas_-_Love_of_My_Life.mp3
Color_Out_-_Alone.mp3
Infraction_-_Sensitive.mp3
Tab_-_Old_Money__feat._Jade_Gritty__amp__AURC_.mp3
Iris_Bevy_-_Bright_Red_and_March_Away.mp3
makesound_-_Inspiring_Cinematic_Piano__Version_1_.mp3
Smoking_With_Poets_-_to__Aurora.mp3
```

```
~/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 >>> █
```

HOW IT WORKS

> python2 get-database-stat.py

it is a python script used to get the names/details of the songs whose fingerprint is created and stored in the database.

```
~/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 >>> █
```

HOW IT WORKS

> `python2 recognize-from-microphone.py -s n`

It is a python script used to get audio input from microphone and returns the name of the song being played

Options

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

`n` – number of seconds

Example: `python2 recognize-from-microphone.py -s 10`

Will listen for 10 seconds

```
~/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:

An 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 APPS

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,

1. Shazam
2. SoundHound
3. Audiggle
4. Audio Tag

And many more..

PREVIOUSLY DEVELOPED MIR APPS

1. Shazam

Shazam is an application that can identify music, movies, advertising, and television shows, based on a short sample 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

PREVIOUSLY DEVELOPED MIR APPS

2. 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

OPEN SOURCE ALGORITHMS

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

1. **Essentia**

C++ library for audio and music analysis, description and synthesis, including Python bindings.
<http://essentia.upf.edu/>

2. **Musicinformationretrieval.com**

Instructional notebooks on music information retrieval.
<http://musicinformationretrieval.com/>

3. **Meyda**

Audio feature extraction for JavaScript.
<https://github.com/meyda/meyda>

OPEN SOURCE ALGORITHMS

4. Madmom

Python audio and music signal processing library

<https://madmom.readthedocs.io/>

5. Strugatzki

Algorithms for matching audio file similarities

<https://github.com/Sciss/Strugatzki>

6. 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>

AUDIO-FINGERPRINT-PYTHON PROJECT

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**.

REFERENCES:

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

<https://musicinformationretrieval.com/index.html>

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

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.

Jagadeesh C (18bcs033)

Anuj C S (18bcs009)

Arun Kumar S M (18bcs013)

Darshan R P (18bcs023)