

MICA: Manual

Music Identifying and Classifying Application

by

Jaxi Li
Weichen Zhang
Yuan Fang

COMP5425
Multimedia Retrieval
Project Final

Contents

1	Introduction	1
2	Installation	2
2.1	Python	2
2.1.1	Mac OS X	2
2.1.2	Ubuntu	2
2.2	Dependencies	2
2.2.1	pygame	2
2.2.2	Others for Processing Audio and Data.	3
3	How to Run	4
3.1	Instrument Retrieval	4
3.1.1	Steps.	4
3.1.2	Training Dataset.	5
3.2	Music Retrieval.	5
3.2.1	Steps.	5
3.2.2	Music Database	6
3.3	Rating Plays.	6

1

Introduction

This manual contains detailed steps for setting up environment for running this application in personal computers. First of all, building a workable environment by installing dependencies is given in the Chapter 2. What follows is guidelines of operating this specific application in Chapter 3.

2

Installation

This whole application is written in *Python*, therefore, there are few crucial *python* dependencies have to be installed before running this application properly. Besides, this application is built and tested under Unix OS (Mac, Linux etc.) only. If you are using Windows, please switch to Linux or Mac using virtual machine or dual boot.

2.1. Python

The version of *Python* using in this application is 2.7.

2.1.1. Mac OS X

Install Python2.7 with following command if you have Homebrew¹:

```
$ brew install python
```

Then, install *pip* for installing Python packages:

```
$ sudo easy_install pip
```

Check if succeed using:

```
$ pip --version  
> pip 8.1.2
```

2.1.2. Ubuntu

Ubuntu OS came with Python interpreters. Therefore, make sure you have installed python development tool and *pip*:

```
$ sudo apt-get install python-dev python-pip
```

Check if succeed using:

```
$ pip --version  
> pip 8.1.2
```

2.2. Dependencies

2.2.1. pygame

pygame is used to build GUIs. Install with *pip*:

¹<http://brew.sh>

```
$ pip install pygame
```

Make sure you can import without errors as following:

```
$ python  
>>> import pygame  
>>>
```

2.2.2. Others for Processing Audio and Data

```
$ pip install scipy numpy pandas pillow sklearn pydub
```

If errors produced, try run with *sudo*. If no errors produced, we are ready to go. See Chapter 3.

3

How to Run

Start the application from command line (Terminal), direct to the folder containing *gui.py*. Run:

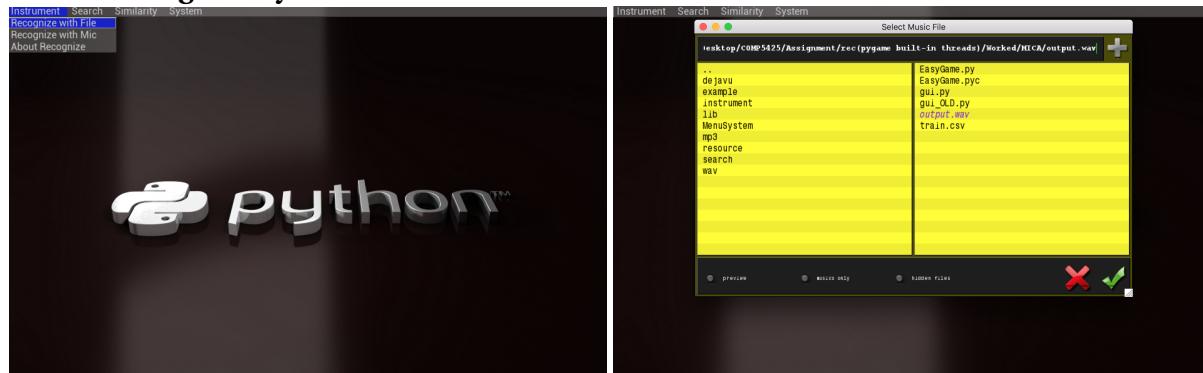
```
$ python gui.py
```

3.1. Instrument Retrieval

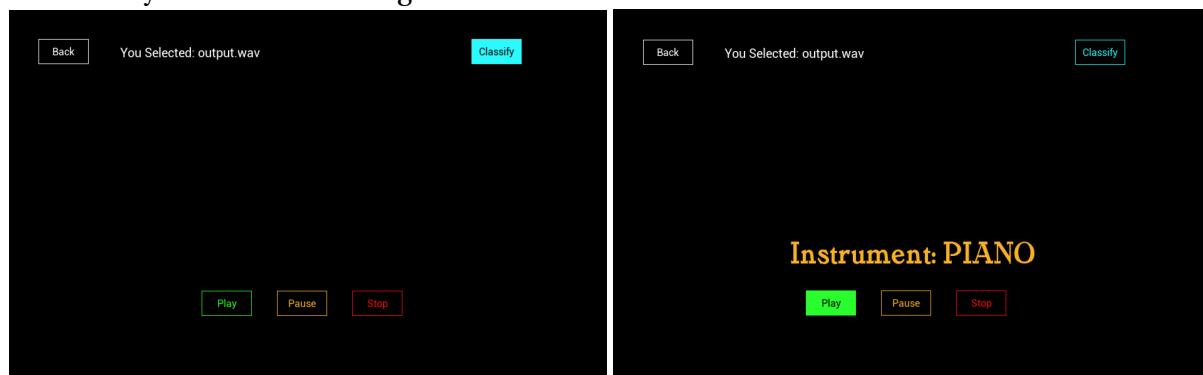
3.1.1. Steps

Once GUI shows up in a new window, select task you willing to proceed, "Recognize by File" or "Recognize by Mic".

Case 1: Recognize by File.

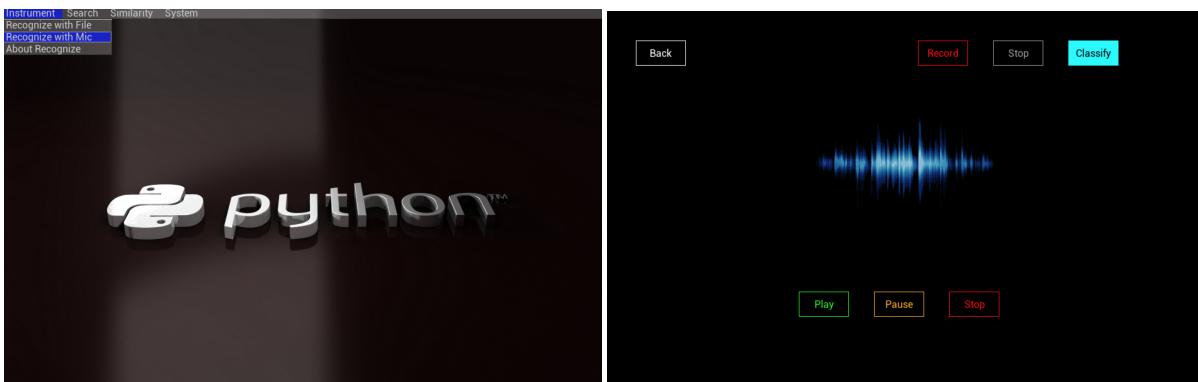


Select file you wish to be recognized.



Click on "Classify" button to let system detect instrument in selected audio file.

Case 2: Recognize by Mic.



Click on "Record" button to record a clip of audio.



Click on "Classify" button to let system detect instrument in recorded audio clip.

3.1.2. Training Dataset

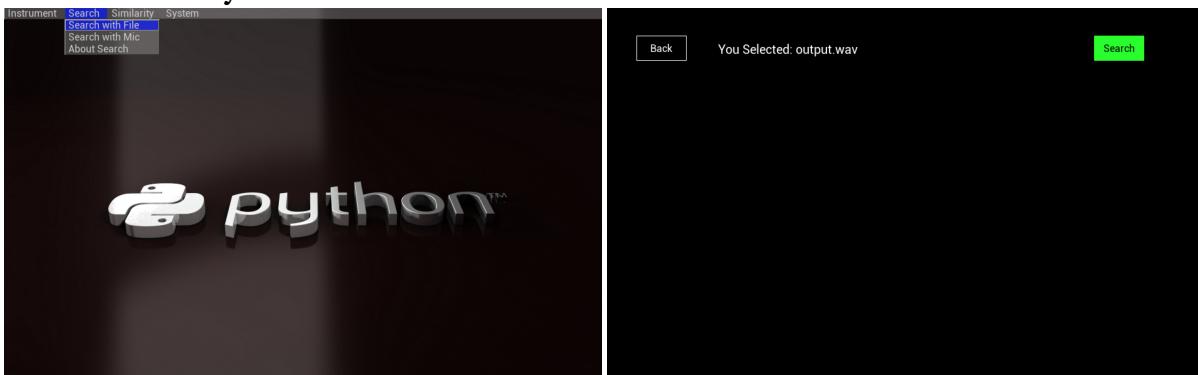
The training dataset (*train.csv*) is generated through *gen_train.py* under "Instrument" folder.

3.2. Music Retrieval

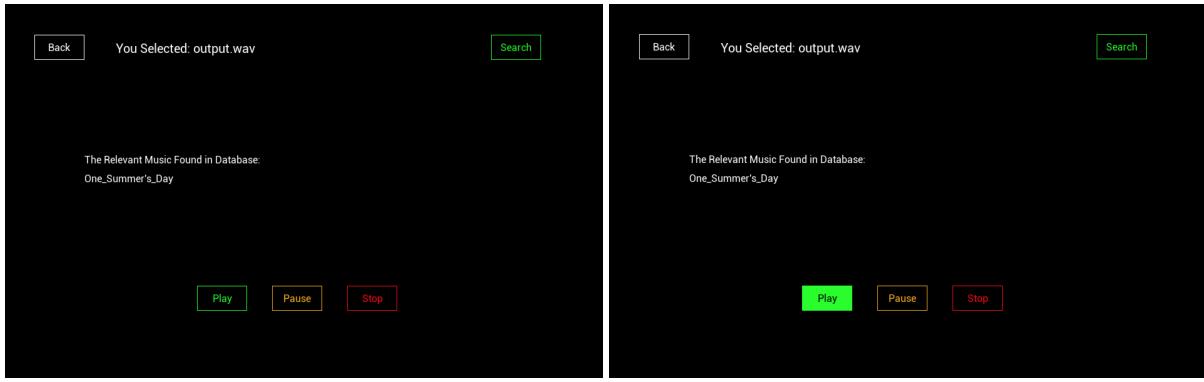
3.2.1. Steps

Select task you willing to proceed, "Search by File" or "Search by Mic".

Case 1: Search by File.

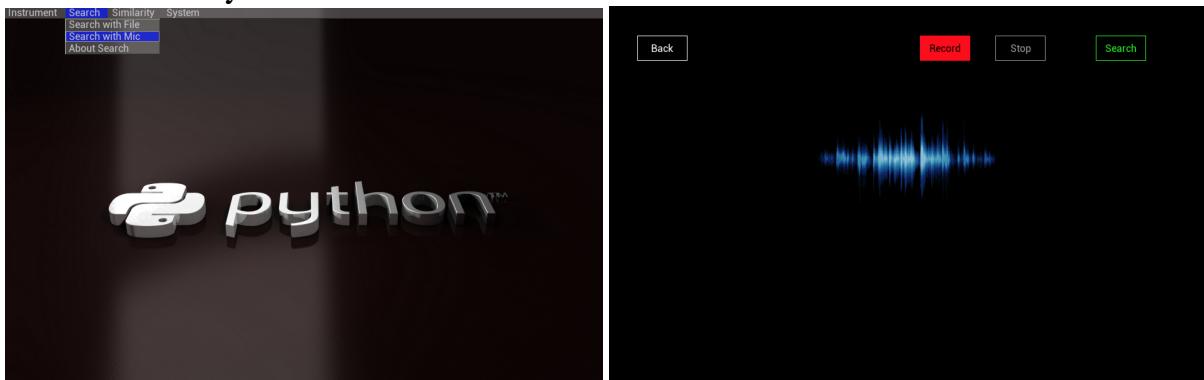


Select unknown file you wish to be matched in database.

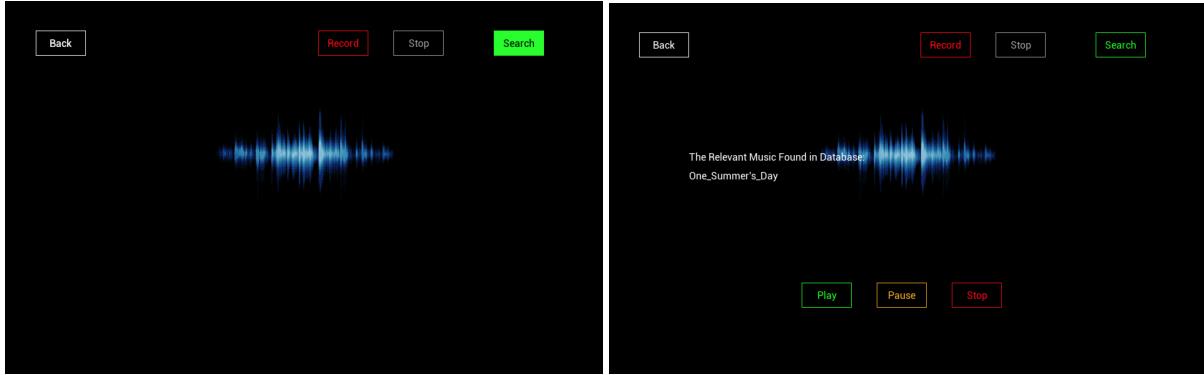


Click on "Search" button to let system detect relevant match to selected audio file.

Case 2: Search by Mic.



Click on "Record" button to record a clip of audio.



Click on "Search" button to let system detect relevant match to recorded audio clip.

3.2.2. Music Database

The music database is located in "wav" folder under "core" directory.

3.3. Rating Plays

TODO