CS4347 Assignment 1
Note-level Singing Voice Transcription

Semester 1, AY2022/2023

(Note: This instruction aims to let you familiarize what knowledge will be required for the assignments. The final version of the instruction will be released at the official start date of this assignment.)

1 Overview

Welcome to the course CS 4347! In the first assignment, you will be instructed on how to do automatic music transcription. Specifically, you will write codes to implement the note-level singing voice transcription. The purposes of this assignment are to learn the basic preprocessing skills of singing data, to implement a basic deep-learning-based transcription framework, and to finish a note-level singing voice transcription system.

References: Lecture 2, 3, 4, AMT Overview, Evaluation for Singing Transcription

Honor Code. This assignment constitutes **15%** of your final grade. Note that plagiarism will not be condoned. You may discuss the questions with your classmates or search on the internet for references, but you MUST NOT submit your code/report that is copied directly from other sources!

2 Submission Instructions

Items to be submitted include:

- **Source code**
  - **dataset.py** This is where you fill in all your code.
  - **main.py** This is where you fill in all your code.
- **Report** The report needs to be named as **report.pdf**.

Please zip the all files together and name it in the following format:

A1234567X_Name_Assignment2.zip

Please submit your assignment by **Sunday, 4 Sep 2022, 2359HRS** to LumiNUS. 25% of the total score will be deducted for each day of late submission.

3 Getting Started

You need to install certain python packages before starting your implementation. Run the following command:

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

If you have any issues with the installation, please post them in the forum so that other students or the instructors can help accordingly.

4 Singing Melody Transcription

Automatic note-level singing voice transcription is an important task in Music Information Retrieval(MIR) community. The objective of the task is to transcribe the singing voice melody into sheet music with onset, offset and pitch information of each note. To complete the singing voice transcription system, you are expected to follow the steps:

- Singing data preprocessing
- Model training using processed singing data
- Transcription evaluation

Each step is detailed in the following session.

4.1 Singing Data Preprocessing [4 points]

You are provided with a singing dataset with 100 pop songs and annotations of each song, where 80 songs are for training, 10 songs are for validation, and 10 songs are for testing. Each song is a mixture of vocal and background music.

Before transcribing the singing voice, you first need to process the raw audios in the given dataset. A simple and common strategy is taken and detailed in the following steps:

1) Load the audio data and re-sample them with a sample rate = 44100 Hz
2) Compute Constant-Q Transform of the audio data
3) Segment all samples into frames with 1024 samples as one frame where the time of the i-th frame is $i * \frac{1024}{44100}$

4) Determine whether each frame contains one note and whether the frame contains the onset or offset of the note. The label of one frame should be in the format [is_onset, is_offset, octave_class_number, pitch_class_number], where is_onset and is_offset are binary. The octave_class_number (5 classes) is in the range of [0, 4], where classes from 0 to 3 correspond to four octaves from 2 to 5 (only consider notes ranging from C2 (note number 36) to B5 (note number 83)) and class 4 corresponds to one unknown octave (e.g., Silence). The pitch_class_number (13 classes) is in the range [0, 12], where classes from 0 to 11 correspond to twelve pitch names from C to B, and class 12 corresponds to one unknown pitch name (e.g., Silence). For example, a frame with the label [1, 0, 1, 0] means this frame has an onset but no offset of the note C3 (note number 48). Please finish the code of this step in function *get_labels* of *dataset.py*.

5) Pad each frame with its context frames, left n frames, and right n frames (if no context, pad zeros) so that continuous $2n+1$ frames will be used as one input data point with shape $(2n+1, \text{feat\_dim})$ for model training

Please finish the code of step **4** in **dataset.py**. **[4 points]**

Hint: Before processing the whole dataset, you are suggested to deal with one song first.

At the end of this step, you will obtain two pickle files, **train.pkl** and **valid.pkl**, which are the processed data file and will be used in the next step.

4.2 Model Training Using Processed Singing Data [2 points]

In this step, a deep convolutional neural network (CNN) is introduced for transcription. You are provided with a base CNN model called BaseNN.

With the processed singing data in the last step, you can train the model to detect the onset, offset and classify notes. Basically, both onset and offset detection are binary classification problems, so the first two outputs of the model are onset probability and offset probability. The pitch detection is a multi-class classification problem, so the 3rd to 7th outputs are used for octave classification, and the remaining 13 outputs are used for pitch classification. Note that the unknown octave and unknown pitch name are predicted when the model has no idea about the frame (e.g., Silence).

Please finish the training script in **main.py** and train the BaseNN model using processed singing data in the last step to get the best model with the least loss value. A screenshot of the training completion is required to be included in the report. **[2 points]**

4.3 Transcription Evaluation [2 points]

With the best-trained model, the final step is to evaluate the transcription results. Specifically, you need to make predictions of the test data (see *inference.py*) and evaluate the predictions (see *evaluate.py*) by metrics COnPOff, COnP, COn (see more details in Evaluation for Singing Transcription).

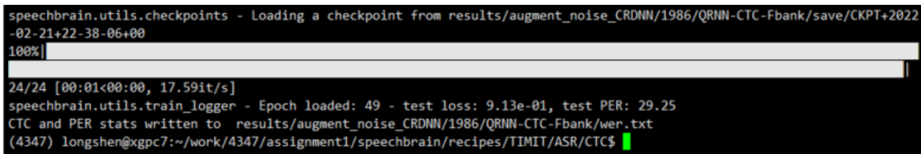
Please run the *inference.py* and *evaluate.py*. The expected F1 result should be more than XX. A screenshot of the evaluation result is required to be included in the report. **[2 points]**

Hint: The inference step also converts the predictions into MIDI files. If you hope to listen to your transcribed results, you can download the MIDI and listen to it. Finally, if the *.mp3* files cannot play well on your machine, you are suggested to use the *utils.py* to convert them to *.wav* files. Then it would be fine to play the *.wav* files.

5 Report [7 points]

At the end of this assignment, you are required to finish a report using the given template. The requirements include:

- Summarize your implementation of each step **[1 point]**
- Result demonstration in Table or Chart **[1 point]**
- Result Analysis **[1 point]**
- Answer the open question: How could the system be improved? **[3 points]**
- Report Length expected to be 2 pages (including figures, tables, not including references) **[1 point]**
- Required screenshots need to contain your username, e.g.,



Figur 1: Screenshot sample

Hint: For the open question, you are encouraged but not limited to consider aspects such as preprocessing strategy, network structure, training process, data attributes, etc. Your answer quality determines your scores on the open question.

6 Bonus [extra 2 points]

Congratulations on finishing Assignment 1! In this assignment, you may notice a case where the onset and offset of a note might occur in the same frame, which may hurt the performance of transcription. You are encouraged to think about how to extend this preprocessing strategy and use a case study to show your extended system is better.

Note that the maximum grade of this assignment is 15 points, which means your final grade for assignment 1 can not exceed 15 points even if you finish the bonus.