

國立清華大學

碩士論文

基於人聲旋律的流行音樂即時伴奏生成系統

**Real-time pop music accompaniment generation according
to vocal melody by deep learning models**

系所別: 資訊系統與應用研究所

學號姓名: 107065525 林子軒 (Tzu-Hsuan Lin)

指導教授: 蘇豐文 博士 (Prof. Von-Wun Soo)

中華民國一一〇年六月

基於人聲旋律的流行音樂即時伴奏生成系統

**Real-time pop music accompaniment generation according
to vocal melody by deep learning models**

Student: Tzu-Hsuan Lin

Advisor: Prof. Von-Wun Soo

July 2021

National Tsing Hua University

Hsinchu, Taiwan, 30013

Submitted in Partial Fulfillment of the Requirements

for Degree of Master in Institute of Information Systems and Applications

摘要

近年來，於現在的音訊處理技術中，我們已經看見在許多技術，如伴奏生成、人聲採譜上都已經取得了一定的成果。然而至今卻還未有將兩者結合的成果出現過，因此，我們融合了目前這些出色的成果，並改進其效率，進而提出了新的「即時伴奏系統」。在這個系統中，我們優化了過去人聲採譜模型的運算效率，以及提出精簡過的HMM-base伴奏生成模型，以在有限的時間內實時生成伴奏。我們認為這個成果將會幫助許多單人歌手獨力創造更完整的表演。

Abstract

The goal of this work is to propose a real-time accompaniment system to assist singers in complete a simple demo by themselves.

By current audio signal technology, we have seen some achievements on accompaniment generation and some on vocal transcription. Basing on these great works, we propose a novel “Real-time accompany generation system” to combine current state-of-arts and further improve the efficiency to reach real-time human interactive mode. To reach a acceptable computing efficiency, we do a lot pruning on original model and apply DenseNet concept to enhance its gradient propagate.

In this system, we integrate efficiency improved vocal transcription model and simplified HMM-base accompaniment generation model which can better fit small training set situation to output musical accompaniment in limited time. We believe that this work will benefit many solo singers to deliver their live shows or demos by themselves whenever they need.

As a result, we reach real-time under 180 BPM which covers most of pop music and propose a highly improved vocal transcription model with 1/1000 parameters and 1/50 FLOPs.

Acknowledgement

First, I want to appreciate my advisor, Prof. Von-Wun Soo. He gave me a lot of advises of experiment design, topic selection and other related issues. Moreover, in the writing stage Prof. Soo did his best effort to check the details of this thesis and helped me complete more solid contents.

Secondly, I want to appreciate the advisor of the paper this work based on, Prof. Li Su and his students. They were all generous to share their acknowledge to me when I was confused by some details of their paper and contacted them for advises.

Thirdly, I want to appreciate my laboratory mates. Without their support, I can hardly complete this work on time. I especially feel grateful to Chen Bo Wei who did me a great favor in the final experiment stage.

Last but not least, I want to appreciate Chi-sheng Wu senior also as my mentor in intern project in MediaTek. He taught me a lot of Deep Learning details and efficient coding skills. Without these skills, I may not be able to complete this complicated work.

Contents

摘要	i
Abstract	ii
Acknowledgement	iii
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Related Work	6
3 Methodology	9
3.1 Vocal Transcription	12
3.2 Rhythms generation out of the vocal melody	20
3.3 Chord prediction in real time from a short period of prior vocal audio sig- nals	21
3.4 Other details	25

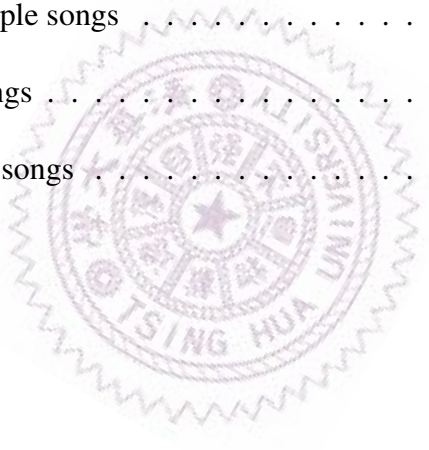


4 Experiments and Results	27
4.1 Real-time feasibility	27
4.2 Dataset	28
4.3 Evaluation	29
5 Conclusion and Future Work	35
5.1 Conclusion	35
5.2 Future Work	36
References	38
.1 Appendix:A	42



List of Tables

4.1	Run time experiment	28
4.2	Performance of the proposed models	30
4.3	Information of sample songs	31
4.4	NCR of sample songs	31
4.5	Hit ratio of sample songs	32



List of Figures

1.1	(left)Spectrum of human voice on C4, (right)Spectrum of guitar on C4. As figures show, left one varies much more than right one.	3
3.1	The overview of the proposed real-time music accompany generation system	11
3.2	The time flow of the proposed real-time music accompany generation system	11
3.3	Difference between baseline model and ours	19
3.4	Example of unstable pitches in attack time and release time	22
3.5	Chord prediction algorithm	25
4.1	Version1: Human composing version, Version2: Model composing version, Version3: Original version; "x" in the box is marked as mean value and dash line in the box is marked as median value. Separated spots are marked as outliers.	33
4.2	We rank the version of highest score as first rank, second highest one as second rank and the lowest one as third rank.	34
1	Parts of our subjective test questionnaire to demonstrate how we conduct our survey	43

Chapter 1

Introduction

Virtual singer is an important technique and we have developed many and mature techniques for singing voice synthesis in our AI lab, which enable our virtual singer to sing a song given its music score and lyrics previously. However, the virtual singer can only passively sing a song according to the music score and cannot interact with human singers if the song score is not given in advance. To extend the virtual singer project to interact with human singers, we need to develop techniques to capture the human singing voices in real time, namely the singing voice transcription. Therefore, we intend to develop a real-time music accompaniment generation system as the first step to enter the interactive virtual singer field. The first interactive singing problem is to design an interactive system that can generate the accompany music to the singer singing in real time. According to the current audio signal technologies, there had been some achievements on accompany generation such as [1] [2] and some on vocal transcription like [3] [4] [5]. However, as far as we know, there is still not yet any practicable applications based on these two techniques, so we are motivated by the thought as how we can combine both works together and deliver a

useful application to the interactive virtual singer applications. That is to develop real time music accompany generation techniques according to the singing voice of a human singer.

Though there have been some great achievements in each aspect, there are still many difficulties to be overcome in order to build a practicable real-time accompaniment generation system. First, as [6] proposed, piano transcription so far can only reach about 90% accuracy. [5] as a state-of-art in the vocal transcription, it shows no more than 80% accuracy and requires much computing resource that is hard to be fulfilled in real-time system. Compared with instrumental transcription, vocal transcription is more complex because of the diverse timbres and unstable pitches in human singing voices. Unstable pitch will make the transcription model misjudge the onset or offset against the nature voice trembles and diverse timbres will increase the complexity of features that result in difficulty for the learning process. Secondly, to reach the real-time performance, we are facing a trade-off in that on one hand, we usually have to clip the input signals into extremely short frames so that we can respond to users in very short delay after one frame has been well processed. However, on the other hand, this method is challenging and almost impracticable for the accompaniment generation system since it can hardly extract enough melody information in less than the half of a bar. Thus, with waiting time and processing time, a noticeable time delay is an inevitable and severe problem to be overcome.

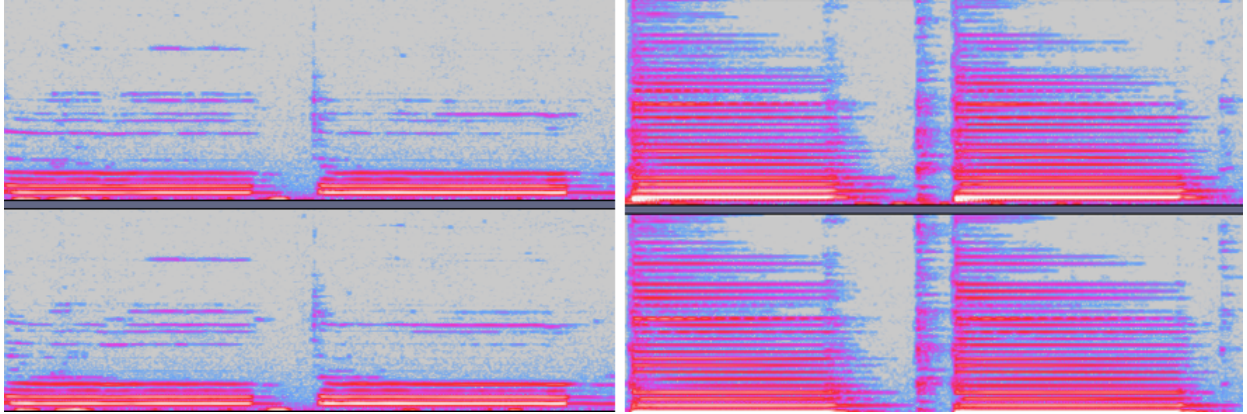


Figure 1.1: (left)Spectrum of human voice on C4, (right)Spectrum of guitar on C4. As figures show, left one varies much more than right one.

To deal with these difficulties, in this dissertation we propose a DNN model structure that can provide better onset detection accuracy. The onset prediction accuracy is a crucial problem in the music accompaniment generation system because it dominates the rhythm of songs and requiring less computing resources. Meanwhile, we build a very light accompaniment generation system including a Markov chain method inspired by [7] to predict future chord flow and an algorithm using onset prediction distribution to detect downbeats of rhythm. With this system, we are able to generate an improvised and reasonable accompaniment with very little computing resources and overcome the delay issue by the prediction modules. The reason why the computing resources reduction is taken into consideration is that we expect that most commercial applications are on the mobile phones which possess much less computing power. Even though we haven't reached this ultimate goal but this work provides a solid baseline toward it.

Here is the summarized goal of our proposal that we are supposed to generate a suitable midi accompaniment with given BPM and tones according to an arbitrary vocal melody in real-time. By real time it means the demanding computing time should be less

than output length of a vocal song at each step. We tentatively set the output length as half a bar. If the output length is too short our generated accompaniment may be very unstable; if it's too long the long delay may cause chord and rhythm predictions difficult and our generated accompaniment will become boring.

We conducted both objective and subjective evaluations for the system. In the objective evaluation, our work performs well with high similarity with human-made works in both harmony and rhythm aspects, which means they're very similar to the music accompaniment created by human.

Our main contributions:

1. Lighten the state of art in vocal transcription model to the one with 1/1000 size, 1/50 FLOPs and only 3 percentage accuracy drop.
2. Build a complete real-time accompanying system and provide related objective and subjective experiment results.

In this work, we propose a real-time system which allows the direct audio input from a microphone and generates pleasing accompaniments in midi form. By the generated accompaniment midi-sheets, the virtual singer can easily sing the harmony for human singers. Furthermore, an important module in this system is the real-time vocal transcription that can be considered as the virtual singer's ears. With the ears, the virtual singer can be turned into playing a role in more innovative applications such as an AI duet virtual singer. Thus, in this work, it not only focuses on the real time music accompany generation but also can actually serve as a blueprint for future interactive virtual singer applications.

In the next section, we will briefly summarize the similar works and related techniques

so far. Then, we will go through the whole structure of this work and introduce used techniques. In the experiment section, objective and subjective experiment results with some interesting insights are provided.



Chapter 2

Related Work

As our best acknowledge, except [8], we didn't find other research doing similar work based on vocal melody. In [8], researchers are trying to produce real-time accompaniment with given music score. In our opinion, it's more like a score tracking technique but a accompaniment generation technique. Moreover, in [8], they use STFT technique to extract frequency domain information (spectrum) and propose an algorithm to do pitch detection and score segmentation based on spectrum, but we didn't find accuracy evaluation of vocal transcription. There is only a case report to show it can detect onset position for a specific song. Thus, we prefer to choose [5] as our vocal transcription module base rather than [8].

Then, we are about to discuss some related works about our modules, vocal transcription and accompaniment generation.

- **Vocal transcription**

As [9] mentions, about vocal transcription, since 2005 researchers have presented some useful approaches of melody transcription [10]. At that time, researchers were still dealing

with instrumental monophonic melody and popular algorithm then were using autocorrelation method to figure out the frequency candidate with largest possibility. From 2008, more and more researchers aren't satisfied with algorithms which only focus on time domain information, Short-time Fourier transform(STFT) or high-resolution Fast Fourier Transform (FFT) are enormously applied into their preprocess step like [11] [12]. More recently, researchers start to pull vocal transcription issue into Deep Neural Network (DNN) field. [3] is published in 2016 and use two DNN models as Voice Activity Detection (VAD) and f_0 estimation modules with STFT processed input. In 2019, [5] used Resnet [13] as DNN model and proposed "hierarchical classification" label to further identify different statements of vocal melody. Our work is also inspired by [5] and treats it as our baseline of vocal transcription system.

- **Accompaniment generation**

On the other hand, accompaniment generation technique is much more mature than vocal transcription field we just mentioned. Especially in DNN field, as [14] mentions, people have done much effort to utilize DNN model to learn music features like counterpoint or musical structures. More recently, [3] used reinforcement learning to build human-machine interactively duet system and [2] used transformer to capture more long-term features to generate multi-instrument accompaniment. However, This kind of techniques requires many computing resources to support model inference, which is not acceptable in real-time system. To reduce required computing resources, we found [7] which used HMM model to select proper chord sequences. This work inspires us a basic chord prediction model with low computing resources but we want to use a simpler way to express this process because

we have only limited pop music chord flow data. The detail of how we simplify it into a Markov chain will be illustrate in the next section.



Chapter 3

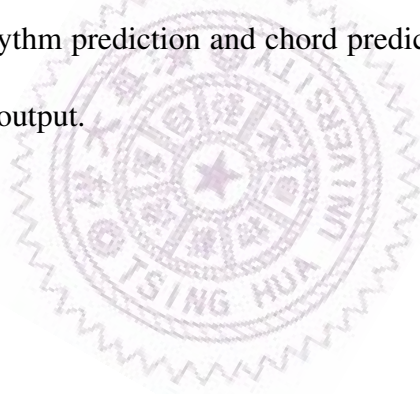
Methodology

We first illustrate the overview flow of the tasks performed in our real-time vocal melody accompany generation system as in Figure 3.1. It consists of three major modules:

1. Vocal transcription
2. Rhythm generation
3. Chord generation.

The vocal transcription module detects the on-off notes from real-time audio vocal signals that are used by the rhythm generation module and the chord generation module to generate the rhythms and predict the chords respectively. The real time input is the original real time audio signals of the vocal melody. It is processed by an combined frequency and periodicity (CFP) approach [15] which use not only spectrum but cepstrum to analyze vocal signal in both frequency and periodicity sides. Beside this method it also applies short time Fourier transform(STFT) and a deep learning Resnet18 model as the music transcription model to predict the on/off notes in the original vocal audio signals.

To explain the definition of real time processing, the time flow of the input data processing is illustrated in Figure 3.2. It shows that the half a bar is processed in one step to generate the output. We allow the system to "listen to" one bar vocal signal before generating an accompaniment output with half a bar length. It means if we can generate an output within the time of half a bar, we can continuously output accompaniment in real time. Because the first output is based on the input signal at Bar 2.1 while the next output is based on the input signal at Bar 2.2, there is no loss of the data as shown in Figure 3.2. So the processing time for the system to generate accompany can not exceed the time bound of half a bar to achieve the real time goal. within this time bound, the system must complete the tasks of transcription, rhythm prediction and chord prediction in order to generate the complete accompany music output.



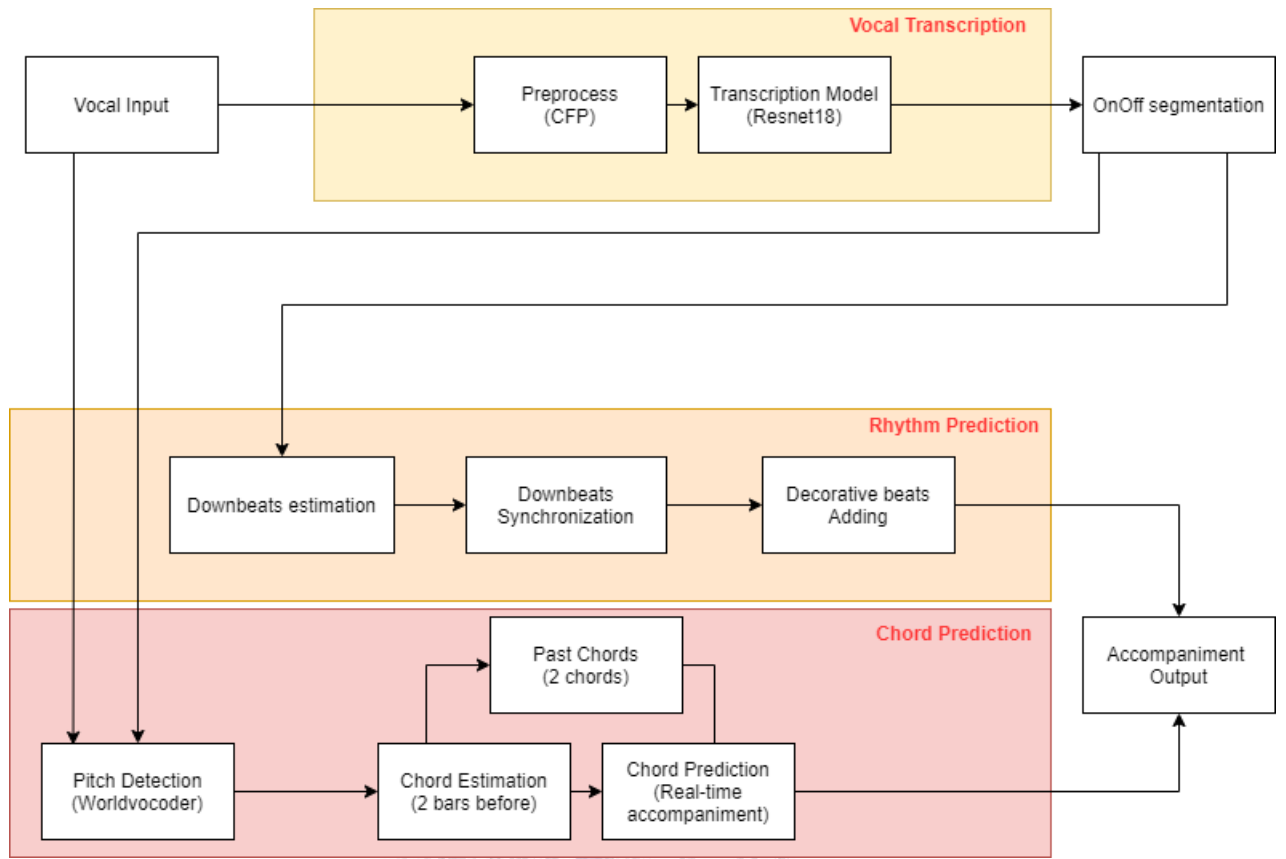


Figure 3.1: The overview of the proposed real-time music accompany generation system



Figure 3.2: The time flow of the proposed real-time music accompany generation system

We then describe each module in detail in the section 3.1, 3.2 and 3.3 respectively.

3.1 Vocal Transcription

- **Label annotation**

In order to transcribe the singing voice melody of a singer, we must extract the audio signals and turn them into their corresponding symbolic music notes correctly. Following the representation of [5], we apply the "Hierarchical classification model" which denotes an audio signal as S, A, O, \bar{O} , X, \bar{X} , and T as the silence, activation, onset, non-onset, offset, non-offset, and transition, respectively to annotate an voice audio signal. Using this representation, we can easily build a loss function for different annotation labels and improve the classification accuracy.

S and A are a set of label to represent if there is a obvious active vocal sound in the given signal frame; O and \bar{O} are also a label set to represent if there is a onset of a note in the given signal frame. And, X and \bar{X} are the one for offset of a note.

- **Data representation**

In this part, we will introduce how our input data looks like and how we process it before sending to our model. The CFP feature extraction method in pre-preprocess mainly follow the same one used in [5].

STFT

First, we begin with the Fourier transform applications in the audio processing domain. Generally, the audio input is recorded as the time domain signal that is known as the wave-form. However, this kind of representation only expresses the magnitude variation that is quite complicated to analyze the detailed information of signals especially when we

are interested in its frequency condition. Fortunately, the Fourier transform allows us depict the magnitude condition in different frequency bands by the Formula 3.1.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{kn}{N}}$$

(3.1)

As a Fourier transform equation, X_k denotes the spectrum signal; x_n denotes the original signal frame; N is the number of original signal frame; k denotes the index of spectrum frequency bands.

In this formula, we can observe the frequency distribution of the signals, but it's not sufficient for us to analyze the audio time domain. In this situation, we can obtain an averaged distribution of the whole signal instead of the variation of frequency with time shifting. Thus, we need to further divide the whole signal into little frames which are short enough to represent variation with time shifting and long enough to provide the needed frequency resolution. Next step, we apply a windowing method and the discrete Fourier transform (DFT) to each frame to obtain the local frequency distributions as Formula 3.2.

$$STFT \{x_n\} (m, \omega) \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x_n w(n - m) e^{-in\omega}$$

(3.2)

This formula is based on DFT and the only new things are m denoting our hop size, w denoting the window function and ω denoting the angular frequency.

In this thesis, we use three different window sizes, “743, 372, 186”, which will be introduced in the next section and a fixed hop size 320 as our hyper-parameters.

CFP feature extraction

In the pre-processing stage, we apply two different feature extraction methods to enhance the information of magnitude variation and fundamental frequency (f_0). To enhance the information of magnitude variation, based on [16], we treat spectrum difference as a better representation to show the magnitude variation which is crucial when we are trying to detect onset and offset positions. In the Formula 3.3, we regard k as the index of spectrum frequency bands, n as the index of time, and X as magnitude part of STFT. The forward spectral difference S^+ and the backward spectral difference S^- are the time-forward and the time-backward respectively.

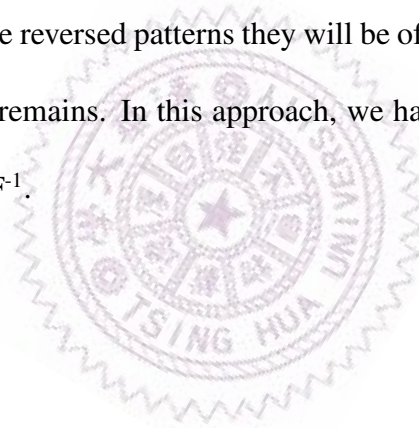
$$S^+ = ReLU(X[k, n + 1] - X[k, n - 1])$$

$$S^- = ReLU(X[k, n - 1] - X[k, n + 1])$$

(3.3)

$ReLU(\cdot)$ denotes the element-wise rectified linear unit. By $ReLU(\cdot)$, S^+ only shows the increasing magnitude part and set all decreasing magnitude part as zero, vice versa.

The combined frequency and periodicity (CFT) approach [15] we used will enhance the fundamental frequency and suppress the subharmonics (f_0/n) and harmonics (nf_0). This approach will combine both the frequency-domain information and the time-domain information. By multiplying them together, because the harmonics and the sub-harmonics in different domains show the reversed patterns they will be offset by each other in this step and fundamental frequency remains. In this approach, we have an N-point DFT matrix F and an inverse DFT matrix F^{-1} .



$$\begin{aligned}
 Z_0[k, n] &:= \sigma_0(W_f X) \\
 Z_1[q, n] &:= \sigma_1(W_t F^{-1} Z_0) \\
 Z_2[k, n] &:= \sigma_2(W_f F Z_1)
 \end{aligned}
 \tag{3.4}$$

where the high-pass filters W_f and W_t in frequency and time domain respectively, and the corresponding activation functions σ_i for each stage is shown as Formula 3.5. X denotes the input signal; $Z_i[k,n]$ denotes the transformed signal from index k .

$$\sigma_i(Z) = |\text{relu}(Z)|^{\gamma_i}, i = 0, 1, 2$$

(3.5)

where $0 \leq \gamma_i \leq 1$

Note that we need to map Z_1 to the frequency domain because it's in the frequency domain in the formulas. To deal with it, we follow [5] to apply two sets of filter banks with 174 triangular filters ranging from 80 Hz to 1000 Hz and there are 48 bands per octave, respectively in the time and frequency domains. Moreover, we use 3 different sample sizes, 186, 372, and 743 of the Hann windows to increase the input feature resolution as recommended in [17]. In Formula 3.6, we denote the filtered Z_1, Z_2 in log-scale as \hat{Z}_1, \hat{Z}_2 .

$$Z[p, n] = \hat{Z}_1[p, n] * \hat{Z}_2[p, n]$$

(3.6)

where $0 \leq \gamma_i \leq 1$; p denotes the index of log-frequency bands; $*$ is an element-dot operation. The equations 3.6 exactly show the CFP method we apply.

- **Data augmentation**

To overcome the limitation of a small dataset size, we did two kinds of data augmentations

before training.

First, we use the synthesis module of "worldvocoder"[18] to generate different vocal versions with one or two semi-steps of a higher or lower key. In other words, each training vocal clip will become five different versions, origin, 1-key higher, 2-key higher, 1-key lower, 2-key lower in the end. This step enriched our dataset and effectively increase our performance.

The reason why only use two semi-step key augmentation is the limitation of synthesis module of "worldvocoder". By our human listening check, key augmentation will bring our data inevitable quality deterioration and it will aggravate along with the increase in the augmentation intensity and the two semi-step key augmentation is the worst version we can still recognize the original melody. Thus, we think it will be noise in our dataset if we apply the key augmentation over this range.

Secondly, we apply volume augmentation in our training process. In every training step, before we start training we will randomly scale the magnitude of the whole spectrum. This augmentation mathematically equals to directly scaling the volume of the time domain signal since STFT function is a linear function. Thus, the process will looks like .

$$s * STFT(x) = STFT(s * x)$$

where $s \in \mathbb{R}$ denotes the scale parameter we apply to the raw signal.

By this volume augmentation, we not only mitigate the overfitting problem caused by the small size but also enhance the robustness to volume variation of this model.

- **The Deep Learning Model**

In the model structure, we lighten Resnet [13] used in [5] mainly by reducing the number of layers and replacing some skip-connection by the connections proposed in Densenet[19] and further speeding it up with minimum performance loss. Meanwhile, we attempt to apply Resnest[20] to enhance the accuracy with almost the same parameters, but it causes slower computing speed resulted from split the channel structure.

Resnet

When researchers try to increase the layers in DNN model to extract more high level features, it's a common phenomenon that the gradient may become uncontrollable, either exploding or vanishing. To solve this problem, [13] proposed a residual adding method to mitigate it.

Lightening the model

The strategy to lighten this model is to reduce the parameters from the deepest and the most time-consuming part. By Pytorch analysis tool, we found that the fourth layer in original model costs over 20% computing resources and the training loss will constantly drop when the training is going even the performance may already be achieved their bound. Thus, the two steps we took are removing redundant layers from our model and reducing the channel number in each layer to eliminate the model parameters. Meanwhile, to improve the training efficiency and the effectiveness of the gradient propagation, we applied the dense structure between layers as in [19] . In Figure 3.3, it shows, we enhance the connection within layers to make gradient propagation more effective and reduce the number of channels from the beginning to eliminate the computation cost.

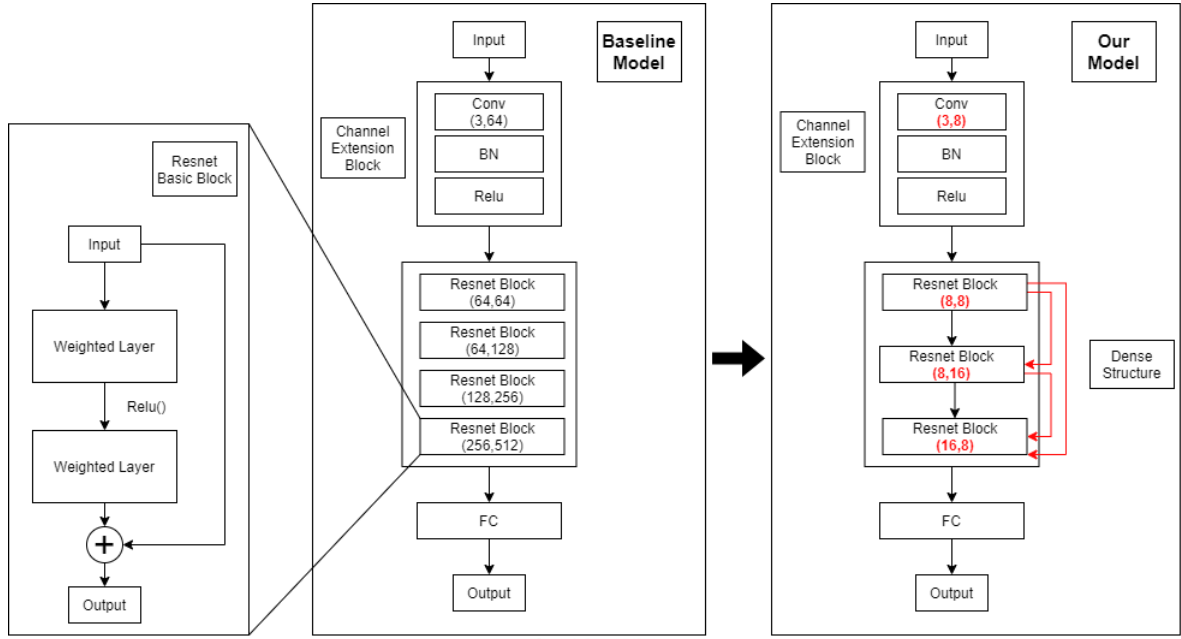


Figure 3.3: Difference between baseline model and ours

Loss function

We use the sum of binary cross entropy of the active statement, the transition statement, the onset statement and the offset statement as the loss function to optimize the model variables as "HCN2" method in [5] as Formula 3.7 shows. One more thing worth mentioning is $BCE(y_{tri}, \hat{y}_{tri})$ put here is an additional term to make transition label more important to model.

$$\begin{aligned}
 LOSS(y, \hat{y}) := & BCE(y_{tri}, \hat{y}_{tri}) + BCE(y_{act}, \hat{y}_{act}) \\
 & + BCE(y_{on}, \hat{y}_{on}) + BCE(y_{off}, \hat{y}_{off})
 \end{aligned}$$

(3.7)

where BCE denotes binary cross entropy function; y_{tri} , y_{act} , y_{on} , y_{off} are further labels show if there is a transition, active vocal sound, onset or offset statement.

3.2 Rhythms generation out of the vocal melody

We know the stable drum patterns can in general be considered as the basic rhythm in the pop music. However, it can be quite obscure or missing in the vocal melody in practice. Thus, we propose the method of generating proper rhythms basing merely on vocal melody. By vocal transcription we have done previously time segments of notes and onset distribution are provided by our model. Because strict downbeats detection is a very complex work and require a lot computing resource as in [21], to reduce computing time we apply a compromised method to detect downbeats. To detect positions of downbeats in short time, we use the peaks of onset distribution from our vocal transcription model because they generally appear when the magnitude drastically increases, which is a typical feature of downbeats and we use this feature to simulate downbeats in this work. Although this is not a precise definition of downbeats, it still provide an similar result with human-made accompaniment in our experiment. In this work, to avoid predicting unstable rhythms, we only detect the downbeat with the highest probability in each bar.

By onset peak detection, we can find a specific onset time which shows the downbeat of vocal melody. Then, because the temporal uncertainty of the vocal melody that may come from amateur singer, we need to stick the downbeat to the eighth note to stabilize the tempo. The reason why we can use the downbeat position in the last bar as the current

accompaniment rhythm is the fact that the pop music rhythm usually stay unchanged in several bars to prevent a listener from being interfered. Thus, even if there is a delay for one bar, our rhythm prediction is still reliable in most of bars.

Beside the downbeat, we need other decorative beats to enrich our rhythm. First, we always set the first beat as bass beat where will be a complete chord with a one octave lower root note. Secondly, to complement the melody bar with relatively few notes, we add decorative beats on 2^{ed} and 3^{rd} if the detected downbeat is in the second half of the bar and 7^{th} and 8^{th} if the detected downbeat is in the first half of the bar. By this post-adjust, our accompaniment will better complement the vocal melody when it comes to a part with less variety.

3.3 Chord prediction in real time from a short period of prior vocal audio signals

- **Pitch Detection**

Pitch detection is another critical module. Though there have been many great accomplishments by DNN methods, they are rarely being practicable as the real-time system because of their computational cost. To minimize the cost, we apply a high-efficient audio signal process tool, the worldvocoder [18], and a light algorithm to overcome the pitch detection issue in vocal pitch tracking.

Worldvocoder

”Worldvocoder” is a high speed audio signal synthesis system which can extract fundamental frequency, spectral envelope and aperiodic parameter from wave form signal and reversely synthesize wave form signal from these features. In this work, we use extracted fundamental frequency feature to predict pitches in each note. More specifically, our pitch prediction is based on the fundamental frequency feature extracted from the Worldvocoder. But there are still some post-processing we need to do to obtain stable note pitches. First, we have to map the frequency feature to midi numbers as musical interval. To reach this goal, we can use every 0.1 second as time frames and do ceiling or flooring to fit midi numbers. However, with this direct process, we confronted with very unstable results that are resulted from the attack time and the release time due to human voice is very unstable. By repeated observation, we found 1/2 to 3/4 period in every note is relatively stable and fit the truly listening cognition of audience. Thus, in this work we only use this period to do our estimation.

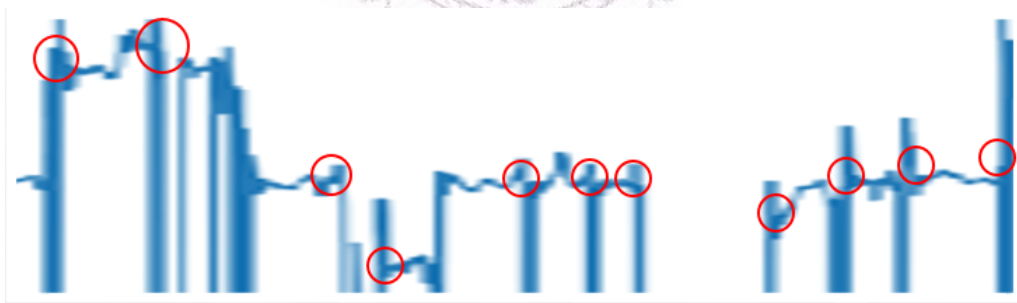


Figure 3.4: Example of unstable pitches in attack time and release time

Secondly, because worldvocoder applies ”DIO” [22] which is based on largest probability of periodical examination on signal to reach fast and relatively stable f_0 estimation and human voice is so unstable that f_0 won’t always possess largest magnitude, we have to smooth some mistaken as higher or lower one octave notes to prevent abrupt pitch leaps.

- **Chord estimation**

In [7], researchers use HMM model to predict the harmonic chord according to given notes. However, in this work, HMM model may not be well trained because of lack of data of pop music lead sheets. As an alternative solution, we simplified the HMM model into a pure transition probability matrix and a chord determination matrix like a Markov chain. To build the transition probability matrix, we collect frequently used chord flows and calculate the occurrence probability of each chord given two previous chords. About chord determination matrix, we simply build a matrix to record the duration time of every notes and select the chord with maximum duration time of the chord-tones. The calculating algorithm is as following shows.

Algorithm 1 Chord estimation

```
1: procedure Chord estimation( $N$ )  $\triangleright N$  denotes the set of all notes, a note with C pitch
   and 0.4 second duration will be record as (0, 0.4)
2:   Initialize chord matrix int  $M[7][7]$   $\triangleright$  To show chord structure, for example we use
   1, 0, 1, 0, 1, 0, 0 to represent C major
3:   Initialize float  $V[7]$ 
4:   for all  $n$  in  $N$  do
5:     for  $i$  in  $range(7)$  do
6:        $V[i] += M[i][n_1] * n_2$   $\triangleright$  Plus the value of note duration, if the note pitch is
   in chord structure
7:     end for
8:   end for
9:   return  $argmax(V)$   $\triangleright$  Select the chord with highest duration sum
10: end procedure
```

Beside the mentioned methodology, because of inevitable time delay we have to predict on chord flow to provide a real-time harmonic accompaniment for the vocal melody. In this, work, we use two previous chords to predict current chord according to common pop music chord flow statistical data. Our prediction process is based on the probability of

all alternative chords given previous two chords to randomly choose the next chord. This is equivalent to the bi-gram approach in the NLP language model. By applying this random process, we diversify our accompaniment and make it more like an improvisation of the chord. We also provide the calculating algorithm as following one.

Algorithm 2 Current chord prediction matrix

```

1: procedure Matrix generation( $S$ )                                ▷  $S$  denotes the set of all chord flows
2:   Initialize  $\text{int}M[7][7][7]$                                   ▷ To record 3-chords flow
3:   for all  $s$  in  $S$  do                                         ▷  $s$  looks like (0,1,6) for C-D-A chord flow
4:      $M[s_1][s_2][s_3] += 1$ 
5:   end for
6:   for  $i$  in  $\text{range}(7)$  do
7:     for  $j$  in  $\text{range}(7)$  do
8:        $M[i][j] /= \text{SUM}(M[i][j])$ 
9:     end for
10:  end for
11:  return  $M$                                                     ▷ Given C-D chord flow,  $M[0][1]$  is the probability of next chord
12: end procedure

```

Moreover, as we show in Figure3.5, because we use half a bar as our input frame size we can use a temporary prediction when we haven't finished processing the whole bar information (Bar n.1 + Bar n.2) and start correcting it when the whole bar information is finished.

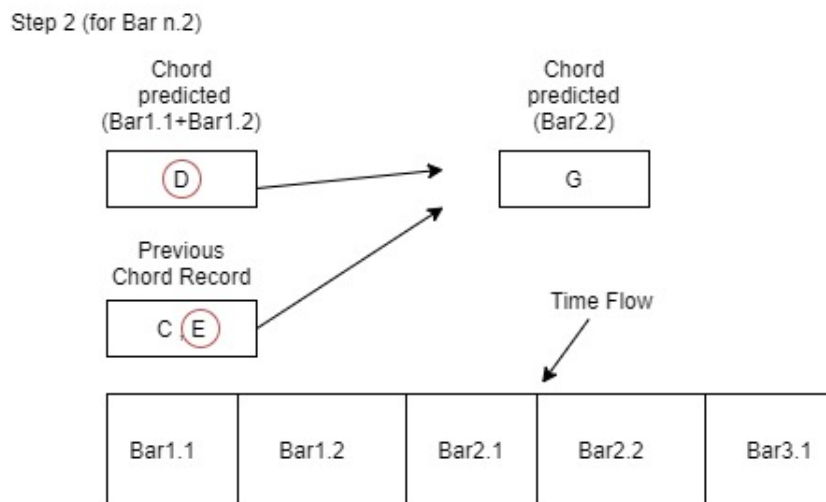
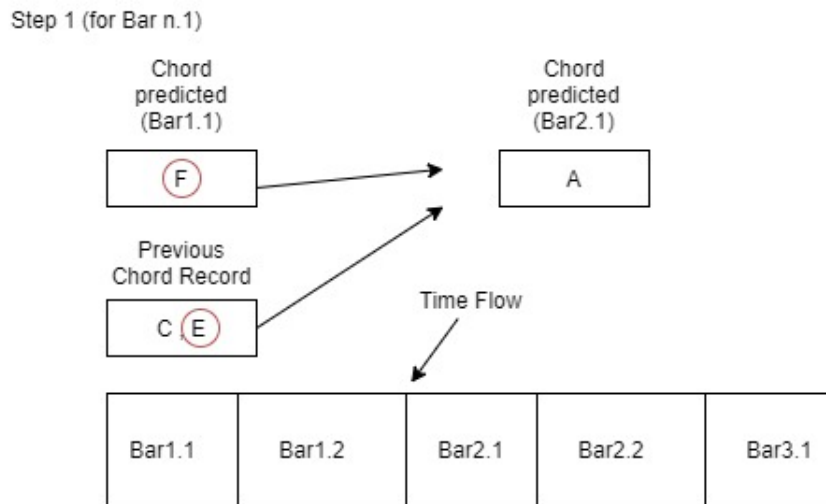


Figure 3.5: Chord prediction algorithm

3.4 Other details

- **Volume adaptation**

To fit a vocal melody better, we also developed volume adaptation module to allow the system automatically adjust the volume of the accompaniment along with a vocal volume

variation. To reach this goal, we set an original volume reference at the beginning when users start to input their voice. In the following bars, system will base on the ratio of volume between the current part and the referenced part to automatically adjust the volume of accompaniment. Without this module, the accompaniment may overwhelm the vocal melody especially in verse part.



Chapter 4

Experiments and Results

4.1 Real-time feasibility

To show a better view of how our system can support real-time operating demand, we have done some run time experiments on our developing platform which is MSI GP75 laptop with i7-9750H CPU and GTX 1660 Ti GPU. Table 4.1 is the result we run our system with direct vocal melody file input, which shows the fact that our computing time is much less than one step duration. In this work, we develop our algorithm to provide one-step long accompaniment by two-steps previous information as we've shown in section 3.3.

In other words, we can reach real-time effect if it's possible to generate output accompaniment within one step duration. Here we set one "step" as a half of a bar. For example, in 60 BPM condition, because one bar duration is 4 seconds if we can complete our computation within a half bar, 2 seconds, we can reach the real-time effect. As you can see, on our

developing platform we can easily reach this requirement in 60, 120 and 180 BPMs which have covered most of existed pop musics. The reason why we choose different BPMs to conduct our experiments is that in addition to the varying computing cost, there is still some fixed cost like I/O, which will result in an uneven proportion between the input length and the computing time. It also means that the real-time effect in a higher BPM is much more challenging. In fact, to deal with songs with an even higher BPM whose computation time is over the time requirement limit in some specific melodies and accompany devices, we can still downgrade its BPM to conduct the accompaniment prediction. Of course, it will make the accompaniment quality downgraded as less harmonic with the vocal melody because we have used a compromised less accurate approach.

Table 4.1: Run time experiment

BPM	Steps	Step duration	Time cost per step
60	137	2	1.1441
120	275	1	0.609
180	413	0.66	0.4431

¹ Step duration, Total time cost, Time cost per step are all recorded in second.

² One step here is regarded as half a bar.

4.2 Dataset

- **Vocal Transcription**

To compare the results with the baseline, we followed [5] to use the datasets, TONAS [23] [24] which consisted of 71 acappella sung melodys, as our training dataset. In addition,

we evaluate the proposed method on the ISMIR2014 song melody dataset [25] as [5] did. In comparison with other generative models, this dataset is relatively small. However, we didn't find obvious overfitting in 30 epochs training using this dataset. Thus, we regard its size being reasonable for this task.

- **Accompaniment Generation**

Since the accompaniment sheets dataset of the pop music is hard to obtain, we tried to use the most popular chord flows as our chord prediction data. The reason why it works is the fact that current pop music almost applies similar chord flows that covered a lot of portion in the pop music, especially in Chinese pop music.

4.3 Evaluation

- **Lighten Model**

Experimental meaning

By applying these methods, we can lighten our model size to 1/1000 from original one and with only 1/50 FLOPs(floating point operations) with only 3 percentage F1 score(note) loss. And this model is also the model we consider best fitting real-time system because of its super efficient tradeoff on computing cost and performance.

Beside the lighten model we mentioned, we also constructed a model with both better performance and less computing cost to show the fact that the modified structure isn't merely a tradeoff but a truly improved model. With our new model, when we set the channel

number as 128 we obtain a model with about 1/8 model size, 60% FLOPs and apparently better performance in terms of F1 score in both onset and note prediction which are what we focus on in this work. The reason why this model can reach better result is due to that the gradient vanishing or explosion issues resulted from enormous parameters are being mitigated, which has been also proved in [19] in comparison with Resnet.

Meanwhile, this experiment also shows the fact that our pitch detection algorithm mentioned in section 3.3 at least provides similar note pitch accuracy in comparison with the DNN based method in [5].

Table 4.2: Performance of the proposed models

Name	Parameters	FLOPs	onset(F1)	offset(F1)	note(F1)
baseline	11198k	188.64M	0.786	0.759	0.594
Lighten(k=128)	1532.8k	102.97M	0.8553	0.6824	0.6279
Lighten(k=32)	268.9k	31.02M	0.8063	0.6749	0.5634
Lighten(k=8)	19.8k	4.29M	0.7967	0.6709	0.566
Lighten(k=4)	5.9k	1.85M	0.6864	0.5538	0.3487

¹ k denotes the number of channels extended in "Channel Extension Block"

- **Accompaniment generation quality**

We conduct both objective and subjective evaluations. To prevent our experiment from being dominated by specific parts, we use the first verse and the first chorus of our songs as samples. In objective one, we ask the singer to sing a specific midi melody in which we have an original human-made accompaniment as a ground truth and check the difference between our prediction and the ground truth.

We choose these three sample songs for this evaluation, and provide relevant information as Table 4.3 shows.

Table 4.3: Information of sample songs

Name	Singer	Gender	Language	BPM	Tonic	Release year
Shape of you	Ed Sheeran	Male	English	96	E major	2017
Goodbye	G.E.M	Female	Chinese	68	G major	2015
Sunny day	Jay Chou	Male	Chinese	69	G major	2003

About chord evaluation, to be honest, it’s very difficult to recognize a chord “correctly” because the same melody may have different acceptable chords that can proposed by even human musicians. Thus, in this work we focus on only harmony between chords and notes. We calculate a “note in chord ratio” (NCR) to represent the harmonious level of our accompaniment.

In Table 4.4, we evaluate NCR in 3 different songs and find that the NCR is very similar to that of the human-made accompaniment as our ground truth in harmony aspect. It is also worth mentioning that in the song “Sunny day” we even have higher NCR in comparison with the ground truth. It is because that the human-made accompaniment usually use a cycle chord flow which may sometimes provide a much more stable but less harmonious accompaniment in the sense of NCR with the vocal melody.

Table 4.4: NCR of sample songs

Name	Ground Truth	Ours
Shape of you	0.5	0.4103
Goodbye	0.6126	0.6036
Sunny day	0.4263	0.538

About rhythm evaluation, as we mentioned in section 3.2, because drum pattern is not an explicit information in the vocal melody and the accompaniment rhythm is relatively free in pop music, we can generate several different accompaniment patterns for the same

melody without any violation sense. Thus, we will evaluate if our downbeats are located on the positions of onsets notes in the music sheets. If most of our downbeats satisfy this requirement, the rhythm we generate will basically fit the vocal melody.

In Table 4.5, we evaluate "Hit ratio" in 3 different songs and find that our "Hit ratio" is also very similar to that of the human-made accompaniment. It is also worth mentioning that in the song "Shape of you" we have a much lower "Hit ratio" in comparison to other songs. It is because it has a high BPM and its rap style makes it relatively difficult for our model to capture the downbeats information.

Table 4.5: Hit ratio of sample songs

Name	Ground Truth	Ours
Shape of you	0.8333	0.758
Goodbye	0.6333	0.6222
Sunny day	0.8804	0.8695

On the subjective evaluation, because the pop music accompaniment is usually composed with multiple instruments such as drum, guitar and bass it's unreasonable to directly compare the original accompaniment with ours. We invite well-educated musician to compose a simple accompaniment with piano as the compared instrument and receive feedback from 38 listeners. In our experience, we will provide a clip without accompaniment in advance. Then, we list three versions of clips which are human composing version, model composing version and original version without label. In other words, listeners don't know which version they are listening. We ask listeners to score three versions of songs respectively after listening 30 seconds clips. About our listeners, We find these people on social media such as Facebook and Instagram. They are with various musical background and join this experiment for nearly no payment.

According to Figure 4.1, we can see the median values are not far from human composing ones'. Though we obviously can't beat human composers now, it still shows that we have generated a comparable work against human composing ones. Here is also one thing worth mentioning that we have a relatively better result in Song 1 which is even better than human composing version. The reason of it is considered as the fact that it's the only one English song and our model is trained by English training dataset, which will provide more accurate transcription and generate more suitable accompaniment on English vocal melody.

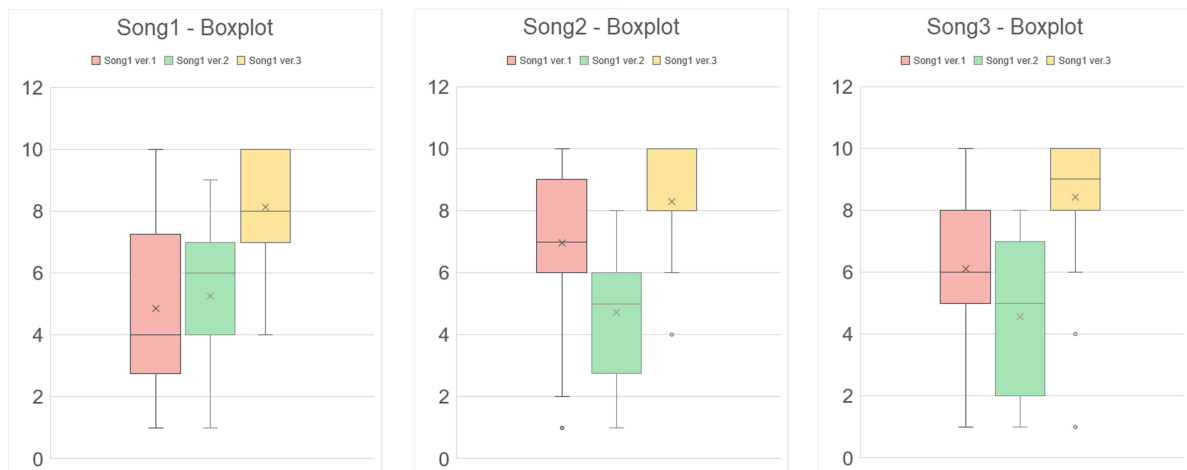


Figure 4.1: Version1: Human composing version, Version2: Model composing version, Version3: Original version; "x" in the box is marked as mean value and dash line in the box is marked as median value. Separated spots are marked as outliers.

About ranking, we can see the fact that even in relatively worse works there are still some listeners prefer our works to human composing ones. By this observation, we think our works may still be able to satisfy listeners in specific occasions.

SUBJECTIVE RANKING

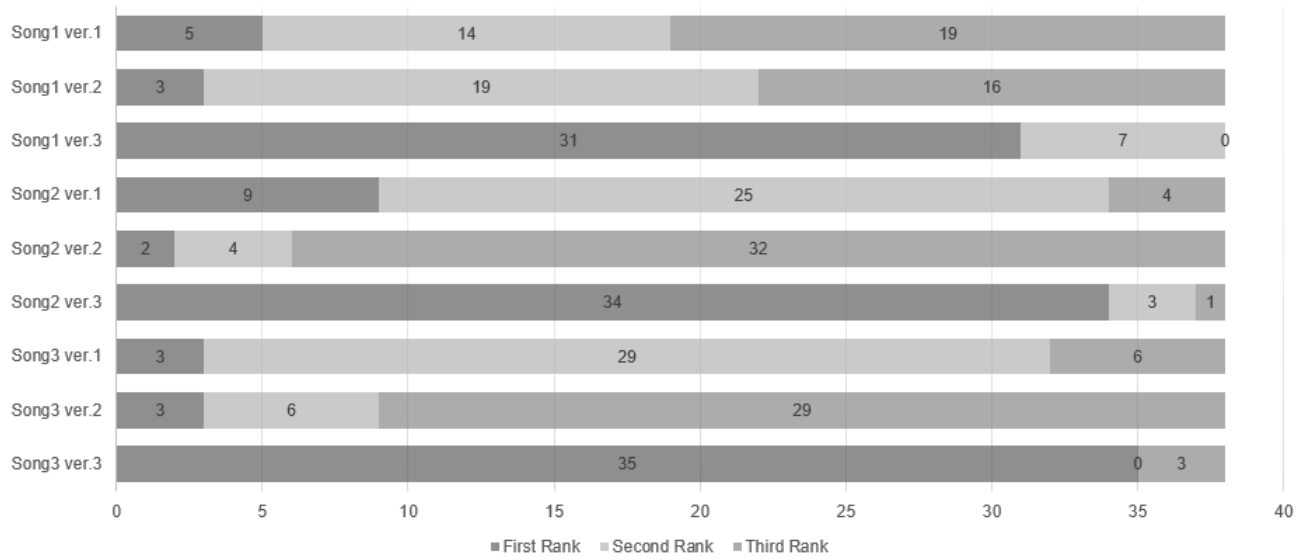


Figure 4.2: We rank the version of highest score as first rank, second highest one as second rank and the lowest one as third rank.



Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this work, we presented a practicable real-time pop music accompaniment generation system which bases on vocal transcription and accompaniment generation techniques. Our contributions are focused on how to integrate these two techniques into a practicable application and how to reduce the computing cost to reach real-time level in as many as possible devices.

First, we developed several light algorithms to generate reasonable accompaniment of given vocal melody such as "Markov chain chord prediction" and "downbeat based rhythm generation". By these algorithms, we can utilize the note on-offset information output from vocal transcription to further construct pleasing accompaniment and overcome inevitable time delay issue we mentioned.

Secondly, we simplified almost all modules to the simplest structure still able to output acceptable result as possible as we can. The proposed algorithms in this work are mainly

based on achievements of previous works and improved by eliminating unnecessary steps to reach faster computing speed and acceptable output. So far, we successfully allow gaming laptop level devices run this system in real-time.

We clearly understand that this work can not perfectly solve general real-time accompaniment generation needs. However, we provided a fundamental and practicable solution for this issue. In our opinion, it won't take too long that our lab can release a commercially practicable version or other related application.

5.2 Future Work

In this work, there are some issues we left because of study time limitation. We will overcome these step by step and deliver a more complete work in the future.

- **BPM and tone automatically detection**

Since these are important parameters in our system and vary between songs from songs, we treat them as needed parameters users have to set manually. By really practice, we found that it's not convenient and user may not be able to provide these parameters if they are lack of musical background. Thus, we would like provide a more user friendly version with this function.

- **Dataset extension**

Though we didn't observe apparent overfitting in current dataset, we are still willing to train this model on larger one. Besides size issue, current dataset is recorded in very high quality which is very difficult to reach in common recording environment. Thus, a lower quality but larger dataset may be a better choice to train our model.

- **Bottom layer API implementation limitation**

To demonstrate our work, we need to play midi file directly to avoid transformation delay so we use "pygame" to reach it. However, "pygame" player doesn't support parallel processing for loading midi and playing midi. Thus, there is always a short blank between bars in real-time playing. Because we think this is only a demonstration in python and will be easily solved when it comes to other platform, we didn't pay a lot effort on solving this issue. About recording, we apply "pyaudio" to help our audio input. To remove unused input in buffer, we will clean it in the first three steps and ensure there is no pre-recorded signals in buffer.



References

- [1] Nan Jiang, Sheng Jin, Zhiyao Duan, and Changshui Zhang. RI-duet: Online music accompaniment generation using deep reinforcement learning, 2020.
- [2] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation, 2020.
- [3] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. pages 1198–1206, 10 2020. doi: 10.1145/3394171.3413721.
- [4] Andrew McLeod, Rodrigo Schramm, Mark Steedman, and Emmanouil Benetos. Automatic transcription of polyphonic vocal music. *Applied Sciences*, 7(12), 2017. ISSN 2076-3417. doi: 10.3390/app7121285. URL <https://www.mdpi.com/2076-3417/7/12/1285>.
- [5] Li Su Zih-Sing Fu. Hierarchical classification networks for singing voice segmentation and transcription. *International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

- [6] Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. High-resolution piano transcription with pedals by regressing onsets and offsets times, 2020.
- [7] Ian Simon, Dan Morris, and Sumit Basu. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 725–734, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580111. doi: 10.1145/1357054.1357169. URL <https://doi.org/10.1145/1357054.1357169>.
- [8] Li Luo, Peng-Fei Lu, and Zeng-Fu Wang. A real-time accompaniment system based on sung voice recognition. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008. doi: 10.1109/ICPR.2008.4761071.
- [9] J. Salamon, E. Gomez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014. doi: 10.1109/MSP.2013.2271648.
- [10] A. Klapuri E. Gómez and B. Meudic. Melody description and extraction in the context of music content processing. *J. New Music Res.*, vol. 32, no. 1, pp. 23–40, 2003.
- [11] M. Ryyänen and A. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Comput. Music J.*, vol. 32, no. 3, pp. 72–86, 2008.
- [12] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2145–2154, 2010. doi: 10.1109/TASL.2010.2042124.

- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [14] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. Deep learning techniques for music generation – a survey, 2019.
- [15] Li Su. Vocal melody extraction using patch-based cnn, 2018.
- [16] Y.-H. Yang C.-Y. Liang, L. Su and H.-M. Lin. Musical offset detection of pitched instruments: The case of violin. *In ISMIR, pages 281–287,*, 2015.
- [17] S. Böck, A. Arzt, F. Krebs, and M. Schedl. Online real-time onset detection with recurrent neural networks. 2012.
- [18] Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884, 2016. doi: 10.1587/transinf.2015EDP7457.
- [19] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.
- [20] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks, 2020.
- [21] Florian Krebs Sebastian Böck and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

- [22] Masanori Morise, Hideki Kawahara, and Haruhiro Katayose. Fast and reliable f_0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. December 2009. AES 35th International Conference: Audio for Games ; Conference date: 11-02-2009 Through 13-02-2009.
- [23] Joaquin Mora, Francisco Gómez, Emilia Gómez, Francisco Javier Borrego, and José Díaz-Báñez. Characterization and melodic similarity of a cappella flamenco cantes. *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, pages 351–356, 01 2010.
- [24] E. Gómez and J. Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2):73–90, 2013. doi: 10.1162/COMJ_a.00180.
- [25] L. J. Tardón I. Barbancho-Perez E. Molina, A. M. Barbancho-Perez. Evaluation framework for automatic singing transcription. 2014.

.1 Appendix:A

伴奏生成主觀測試

本問卷是對不同條件下製作伴奏的主觀測試，希望知道受試者對三種不同版本伴奏的評價。
本測試著重於整體的悅耳程度，不需要在意其他假設(ex:收聽場合、目標聽眾...), 只需給予聽覺上的直覺感受即可。

本測試因人聲部分為額外提取，所以音質上有部分破損，請盡可能不要將音質納入評分。

非常感謝你願意花時間幫助我度過艱難的碩論最後階段XDDD

為了聊表誠意，有留聯絡資訊的抽三杯星巴克，


抽到的我再私訊，看是你要來新竹拿，還是我剛好遇見你再拿給你XDDD

*必填

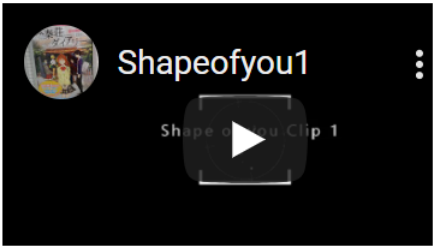
音樂喜好 *

- Pop 流行
- Rock 搖滾
- Folk 民謠
- Electronic 電子
- Jazz 爵士
- Absolute Music 純音樂
- Rap 說唱
- Classical 古典

歌曲一 無伴奏版本



歌曲一 版本1



歌曲一 版本1 *

請以無伴奏版本為零分基礎，對版本1伴奏的悅耳程度打1~10分

1 2 3 4 5 6 7 8 9 10

Figure 1: Parts of our subjective test questionnaire to demonstrate how we conduct our survey