Abstract
Creating a piece of music is difficult for people who have never been trained to compose. We present an automatic music generation system to lower the threshold of creating music. The system takes the user’s humming as input and creates full music based on the humming melody. The system consists of five modules: 1) humming transcription, 2) melody generation, 3) broken chord generation, 4) accompaniment generation, and 5) audio synthesis. The first module transcribes the user’s humming audio to a score, and then the melody generation module composes a complete melody based on the user’s humming melody. After that, the third module will generate a broken chord track to accompany the full melody, and the fourth module will create more accompanying tracks. Finally, the audio synthesis module mixes all the tracks to generate the music. Through the user experiment, our system can generate high-quality music with natural expression based on the user’s humming input.

1 Introduction
Recently, music generation technology has made rapid progress. [Oord et al., 2016; Payne, 2019; Dhariwal et al., 2020] can directly generate music audio, and [Huang et al., 2018; Hsiao et al., 2021; Zhang et al., 2022] can generate music scores. Many works can generate customized music based on user input. For example, [Ju et al., 2021] can generate melody based on lyrics, [Choi et al., 2021] can generate melody based on chords, and [Agostinelli et al., 2023] can generate corresponding music based on a text description. However, compared with lyrics and chord progressions, humming is a more accessible input form for users. (Although it is easier for users to choose mood or music style directly, users’ participation in music generation is limited.)

At present, there are few automatic music generation systems with user humming as input. Some apps, such as HumBeatz¹, HumOn and ZhiQu², only support the accompaniment of users’ humming, cannot automatically complete

¹HumBeatz: https://humbeatz.com/
²ZhiQu: http://zingasong.com/
In this paper, we propose a music generation system that takes the user’s humming as input and generates a piece of complete music based on it. The system consists of five modules: 1) humming transcription, 2) melody generation, 3) broken chord generation, 4) accompaniment generation, and 5) audio synthesis. The humming transcription module contains a deep learning model, which is used to transcribe the user’s humming audio into music scores, and an algorithm to fine-tune some dissonant notes. The melody generation module is to continue the humming melody into a complete melody while ensuring that the humming melody has enough repetition and memory points. The broken chord generation module and the accompaniment generation module will create multiple accompaniment tracks for the melody to improve the richness of the music. Finally, the audio synthesis module will mix all these tracks and add some mixing effects. Through a large number of user tests, our system has been proved to be able to generate music with high quality, and the user humming melody is strongly reflected in the final music.

Therefore, we adopt an approach named MusicYOLO based on the object detection technology in CV (Computer Vision) [Wang et al., 2022]. Firstly, the humming audio is converted into a CQT spectrogram matrix, and then the matrix is converted into a three-channel image by linear intensity mapping. Afterward, the image is fed into the YOLOX model to obtain the note’s bounding boxes. Finally, the boxes will be converted to music notes. The YOLOX used in the paper is only trained on two datasets, SSVD\(^3\) and MIR [Wang and Jang, 2021]. SSVD is only in the sight-sing domain [McClung, 2001], and the MIR dataset’s quality is not high. So, in order to improve the model’s performance and generalization, we further construct a larger transcription dataset based on some SVS (sight voice synthesis) dataset, PopCS [Liu et al., 2022], Opencpop\(^4\) and m4singer [Zhang et al., ]. Besides, we develop an algorithm to fine-tune some dissonant notes to improve the downstream melody generation model’s performance. For example, We set the onset and offset of all notes to be an integral multiple of the length of the 16th note.

2 Methodology

Figure 2 shows the pipeline of the whole system. The technical implementation methods of each module will be described in detail below.

2.1 Humming Transcription

The function of the humming transcription module is to transcribe the user’s humming audio into a music score in midi format. Currently, most transcription models for musical instruments are frame-based. They usually adopt a two-step method to obtain the onset, offset, and pitch of each note [Hawthorne et al., 2017; Kelz et al., 2016; Sigta et al., 2016]. However, these frame-based models fail to work well on human vocals because the onset of the human voice is not as clear and stable as that of musical instruments.

Therefore, we adopt an approach named MusicYOLO based on the object detection technology in CV (Computer Vision) [Wang et al., 2022]. Firstly, the humming audio is converted into a CQT spectrogram matrix, and then the matrix is converted into a three-channel image by linear intensity mapping. Afterward, the image is fed into the YOLOX model to obtain the note’s bounding boxes. Finally, the boxes will be converted to music notes. The YOLOX used in the paper is only trained on two datasets, SSVD\(^3\) and MIR [Wang and Jang, 2021]. SSVD is only in the sight-sing domain [McClung, 2001], and the MIR dataset’s quality is not high. So, in order to improve the model’s performance and generalization, we further construct a larger transcription dataset based on some SVS (sight voice synthesis) dataset, PopCS [Liu et al., 2022], Opencpop\(^4\) and m4singer [Zhang et al., ]. Besides, we develop an algorithm to fine-tune some dissonant notes to improve the downstream melody generation model’s performance. For example, We set the onset and offset of all notes to be an integral multiple of the length of the 16th note.

2.2 Melody Generation

Melody Generation Module is to generate a complete melody based on the user’s humming melody. Although many models can write melodies [Wu et al., 2019; Huang and Yang, 2020], the quality of the generated melody is limited, and the melody quality will decrease as the generation length grows. To solve this problem, we adopt a conditional generation model CMT [Choi et al., 2021], which is similar to the sequence labeling model in NLP (Neural Language Processing). Its input is the sum of the chord and rhythm template representations, and the output is the pitch sequence.

In this module, the transcribed user’s humming melody will first extract suitable chord and rhythm templates from

\(^3\)MIR: https://github.com/itec-hust/Sight-Singing-Vocal-Data
\(^4\)Opencpop: https://wenet.org.cn/opencpop/
a music template library through a template-matching algorithm (The library can be automatically constructed using the algorithm proposed in CMT paper). The method of matching chord templates is to calculate the similarity between the scales of all notes in the humming melody’s first bar and the scales of the first chord’s notes in each chord template (we denote the two sets of scales as \(NS\) and \(CS\)), and select the chord template with the highest similarity score,

\[
Sim_{chord} = \frac{\text{Size}(NS \cap CS)}{\text{Size}(NS \cup CS)} \tag{1}
\]

The method of matching the rhythm template is to calculate the overlap of the onsets between the humming melody and each rhythm template in the first bar (denote the two sets of onsets as \(NO\) and \(RO\)) and select the rhythm template with the highest score.

\[
Sim_{rhythm} = \frac{\text{Size}(NO \cap RO)}{\text{Size}(NO \cup RO)} \tag{2}
\]

In order to make the model able to generate melodies of various styles, moods, and speeds, we have built a large training dataset containing 1 million 8-bar samples (each sample contains one melody and its corresponding chord and rhythm template). The dataset is constructed from POP909 \cite{Wang2020Hookpad}, Hookpad \footnote{Hooktheory: https://www.hooktheory.com/treble}, Wikifonia \cite{Simonetta2018Nottingham}, and Nottingham \footnote{Nottingham Database: https://ifdo.ca/~seymour/nottingham/nottingham.html} Datasets.

### 2.3 Broken Chord Generation

To further enhance the expression and richness of music, we train a broken chord generation model based on transformer \cite{Vaswani2017Attention}. The training idea of the model comes from TeleMelody \cite{Ju2021Telemelody}. The model’s input is the melody generated by the music generation module and its corresponding chord and rhythm template. The output is the note sequence of the broken chord. In order to match the broken chord with the melody and make it strictly implement the chord template, we modify the alignment strategy proposed in TeleMelody to the one shown in Figure 3.

The input sequence contains two types of elements, which represent chords and notes, respectively. Each chord element consists of three tokens, representing the number of bars the chord belongs to, the chord root, and the chord attribute, respectively. Followed by the chord element is the note element. Each note element contains three tokens representing the position, pitch, and duration, respectively. The output sequence contains only one type of element, the note element. Similarly to Telemelody, we introduce musical knowledge to the model through well-designed alignment regularization during training. The pitch-related tokens will be aligned, such as the input sequence’s chord root token, chord attribute token and note pitch token, and the output sequence’s note pitch token. The time-related tokens will be aligned, such as the input sequence’s chord bar token, note position token, note duration token, and the output sequence’s note position token, and duration token.

![Figure 3: Input and output representation and alignment regulation method in the broken chord generation model.](image)

### 2.4 Accompaniment Generation

This module is to generate more accompaniment tracks (bass, string) for music. The implementation method is an expert system based on music theory rules. The method of generating the drum track is randomly selecting patterns from the drum pattern library. Bass and string are pitch instruments, so we first build their pattern library with C as the root tone. After selecting the pattern, we will adjust the root tone of the pattern to that of the chord in the corresponding position.

### 2.5 Audio Synthesis

This module contains three functions. 1) Balance the volume of each track. 2) Add appropriate reverberation for each track. 3) Compressor, which is used to avoid crackling and keep the voice stable, full, and clear. All the above functions are implemented with preset parameters through the python package Pedalboard \footnote{Pedalboard: https://github.com/spotify/pedalboard}.

### 3 Results and Conclusion

Through human evaluation, our proposed Humming2Music system can achieve the following points: 1) In most cases (except when the user’s humming is too fast), the transcription results of user humming are accurate. 2) The melody quality generated by the model is high, and the melody quality will remain the same with the increase of the melody length, mainly due to our use of templates (chord template and rhythm template) to constrain the generation of melody. 3) The humming melody of the user is clearly reflected in the final music. It is because we use the similarity between the humming melody and template from the two dimensions (chord and rhythm) when selecting the template. 4) The generated music is quite different from the template’s corresponding original music. It is mainly because the template only contains the rhythm and chord information. The model generates the most important pitch part. Besides, accompaniment tracks (bass, string, and drum) are generated by randomly selecting patterns from an extensive pattern library. It is almost impossible to be similar to the original music.

According to the questionnaire distributed to users, 46% of users think the generated music is not bad, and 39% of users think the generated music is enjoyable and surprising.
References


