https://doi.org/10.62836/isrm.v2i1.514

Global Science Publishing

ISRM

Intelligent Systems & Robotic Mechanics
https://ojs.sgsci.org/journals/isrm

Article

"Public Acceptance" Rating Model of Classical Music

Yikang Hong

The Experimental High School Attached to Beijing Normal University, Beijing 100032, China

Abstract: A large proportion of classical music is overlooked by the general public but has the potential to become widely accepted. This paper proposes a model to rate the "public acceptance" score, ranging from 0 to 100, of an arbitrary music piece. It serves as an objective, convenient, and effective way of musical analysis. The model involves a standard dataset for comparison, a preprocessing module, a feature extractor, and a rating module. The standard dataset consists of music pieces that have already gained "public acceptance" according to statistics, while the resampling and normalization module preprocesses the audio, and the feature extractor uses pre-trained VGGish to extract vector features of the audio. Then, these features are compared with those in the standard dataset, and the final rating is calculated. In addition, the model has features that make the rating explainable, and it has been proven by objective and subjective evaluation that this model is accurate and reliable, with the ratings on the same piece highly converged and close to that of manual rating. Possible improvements to this model are marked in the end.

Keywords: classical music analysis; audio techniques; feature extraction; VGGish

1. Introduction

Classical music encompasses music work from the medieval age to the early twentieth century. The total amount of known classical music pieces out-ranges 200,000, according to the International Music Score Library Project (Petrucci Music Library) [1]. However, only a small portion of these pieces were sufficiently studied, and a much smaller part is well-known to the general public.

In the field of music recommendation and music analysis, quantifying the "public acceptance" of a new piece is worth studying. Most previous studies focus on behavioral data and the field of music tagging. However, few studies have been conducted on musical features themselves, and fewer of those have focused on classical music. On the other hand, most evaluations of music pieces remain subjective. Thus, an objective rating system is significant for further research.

This paper introduces a model designed to analyze and rate the possible "public acceptance" of music pieces with a relatively low play count, providing a reference for the recommendation function in music software, the selection of works in public concerts, and objective rating for musical research. By uploading the audio form of the music piece into the model, users can get the rating and feedback of the name, composer, and genre of a similar music piece that is considered popular and highly acceptable to the audience according to statistics, which makes the model interpretable.

Still, problems arose. Firstly, training models specifically addressing classical music require abundant tagged data, with basic standards including the genre, form, composer, instruments, tempo, and tonality of a

Received: 4 September 2025; Accepted: 19 September 2025.

Corresponding: Yikang Hong (451789754@qq.com)

piece. While obtaining such accurate data is impossible, creating such a dataset requires far more time and effort. Thus, VGGish [2], a general audio vector feature extractor, seems to be the optimal solution. Input audio is first resampled and normalized, fitting the requirements of VGGish. Then, VGGish extracts vector features from the input. These features were compared with a standard dataset, which consists of the VGGish features of music pieces that are considered “acceptable” accord- ing to view count statistics. Distance between the input audio to the standard dataset is calculated, inferring a final “public acceptance” score.

The main innovations introduced by this project are summarized as follows:

• An objective and explainable evaluation system for classical music. Distinct from existing musical analysis methods, it is not based on music theory or subjective judgment.

• A “high public acceptance” music dataset of audios and VGGish features, based on statistics.

• This model has proved to be accurate, and its ratings achieved a certain degree of recognition among people.

2. Structure

As shown in Figure 1, the entire model consists of four parts: the resampling and normalization module, the feature extractor, a standard dataset, and the rating module. These modules were combined into a continuous model.

• The Resampling and normalization module resamples the MP3 input to a 16 kHz mono audio WAV file. Then, peak normalization is applied to all files, ensuring equality in the peak of sound and preserving the contrast of loudness within one audio.

• The Feature extractor is the general audio vector feature extractor VGGish [2]. It is a pre-trained model based on VGG, a convolutional neural network (CNN) with a deeper depth and a smaller kernel [3]. This model extracts the digital features of the resampled and normalized input.

• The Standard dataset is comprised of VGGish extracted vectors of classical music pieces. These pieces were selected according to their number of views on YouTube, the largest video platform. Music pieces above a certain view count are considered “popular” and added to the dataset. The features were extracted using VGGish after resampling and cutting these audio files, and they were used for comparison with the features of the input.

• The rating module compares the extracted features of the input with the standard dataset one by one, and the closest of these is considered to calculate the final result.

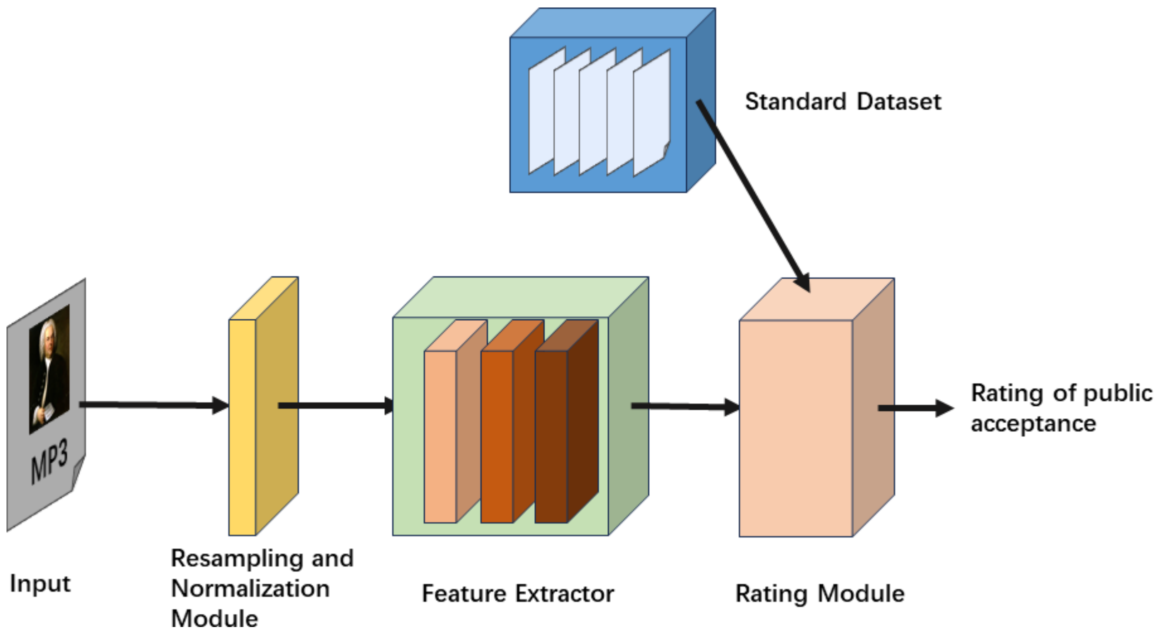
Standard Dataset
MP3
Rating of public acceptance
Input
Resampling and Normalization Module
Feature Extractor
Rating Module

Figure 1. Overall structure of the model.

3. The Resampling and Normalization Module

The resampling and normalization module resamples and normalizes the arbitrary MP3 input into a 16,000 Hz mono WAV file, which meets the requirements of the feature extractor.

Several problems were identified. First, the VGGish [2] feature extractor requires the sample rate of the audio to be 16 kHz, all the audio should be resampled. In this case, fractional resampling is used. Second, different audios, using various recording techniques, have distinguishing loudness. To solve this problem, peak normalization is applied. This process normalizes the peak volume of each audio and preserves the important volume contradiction within the audio.

3.1. Combining Channels

Two channels were combined by taking the average of two-channel values in each frame:

$$\bar{X}[i] = \frac{X_1[i] + X_2[i]}{2}$$

3.2. Fractional Resampling

Because a majority of music files have a sampling rate of $44,100$ Hz, which is not a multiple of $16,000$, fractional resampling is applied to resample the original file. Suppose the simplest fraction between the input sample rate and the target sample rate be $\frac{P}{Q}$.

• Upsampling is done by inserting zero-frames between the original signal so the total amount of frames is Q times the original audio:

$$X_{up}[i] = \begin{cases} \bar{X}\left[\frac{i}{Q}\right] & \text{When } i \bmod Q = 0 \\ 0 & \text{When } i \bmod Q \neq 0 \end{cases}$$

• Low-Pass Filtering is done by applying the Butterworth Filter [4] to eliminate the high- frequency information caused by inserting zero-frames.

• Downsampling is done by collecting information every P frame.

$$X_{down}[i] = X_{filtered}[i \times P]$$

In this process, the peak of the audio, suppose with an amplitude of P, is found. Then, it is modified to a constant, T, by multiplying the fraction $\frac{T}{P}$. All the rest of the frames were modified in the same proportion:

$$X_{res}[i] = X_{up}[i] \times \frac{T}{P}$$

The feature extractor consists of the pre-trained model VGGish [2]. It is based on an optimized convolutional neural network, the VGG Network [3], and it is trained on a large dataset from YouTube. The VGG Network is specialized in image recognition and has specific features, such as a deeper network of 11–19 layers (from 11 layers in Configuration A to 19 layers in Configuration E) and a small 3 × 3 convolutional kernel, that improves its accuracy.

The VGGish model, similarly, is a variant of Configuration A of the VGG model, which has 11 weight layers. In order to process the audio, the model first computes a 96 × 64 log mel spectrogram with a window of 10 ms, making each of these spectrograms cover 0.96 s. This 96 × 64 matrix serves as the input to the VGG model. With a 128-wide fully connected layer instead of a 1000-wide fully connected layer at the end, the output of the VGGish is a 128-dimensional vector for each 0.96 s of audio. After rectification, the vectors have integer values ranging from 0 to 255.

Therefore, the feature matrix of a resampled T-second audio will be $\left\lfloor \frac{T}{0.96} \right\rfloor \times 128$.

4. The Standard Dataset

The standard dataset consists of 245 64 × 128 matrices representing the extracted features of classical music pieces that are already considered acceptable according to statistics. Also, the information on each of these pieces was recorded in a list.

4.1. Collection

Original audio MP3s were collected manually. Specifically, by searching with keywords of the names of the top influential composers [5], music work with total views above 5 million is deduplicated and added to the list, noting that multiple versions of the same work are counted as one single piece. Audio pieces were chosen randomly from various recordings of the same piece, each with two channels and a sample rate of 44.1 kHz.

4.2. Resampling and Normalization

Each raw MP3 file undergoes a similar process as in the resampling and normalization module. First, the two channels are combined into one by averaging the samples across channels. Then, they were resampled with a rate of 16 kHz, followed by a normalization process with which the entire audio is amplified in a certain ratio, so the peak of the signal reaches 1.0. In general, this process produces a normalized 16 kHz mono WAV file out of the original MP3 file.

4.3. Format Unification and Feature Extraction

Music pieces have diverse lengths. To ensure their feature matrices have the same dimensions, a consecutive 61.44 s of audio is randomly selected from each of these WAV files. Then, the VGGish model extracts features of this audio at an interval of 0.96 s, creating a 64 × 128 matrix for each of these audios.

4.4. Compressing Data

The previous 64 × 128 matrix is based on a time sequence. Feature of VGGish [2] model contributed to the irrelevance between the 64 128-dimensional vectors. To eradicate this irrelevance and further compress data, the maximum, minimum, mean, and standard deviation of these vectors were extracted respectively, forming a smaller 512-dimensional vector that represents the entire audio.

Specifically, suppose ai represents the *i*th 128-dimensional vector, we have:

$$\bar{a}[k] = \frac{1}{64}\sum_{i=1}^{64} a_i[k]$$

$$a_{std}[k] = \sqrt{\frac{\sum_{i=1}^{64}(\bar{a}[k]-a_i[k])^2}{64}}$$

$$a_{max}[k] = \frac{1}{64}\max_{i=1}^{64}\ a_i[k]$$

$$a_{min}[k] = \frac{1}{64}\min_{i=1}^{64}\ a_i[k]$$

Then, the 4 vectors were combined, forming a single vector. This vector is used for comparison with a similar vector from the input audio.

5. Rating Module

The rating module compares the features of the input audio with the features in the standard dataset. In order to keep consistent with the format of the standard dataset, 64 consecutive 128-dimension vectors were extracted and gone through the data compression mentioned in Section 4.4.

After obtaining the 512-dimensional vector, the Euclidean metric of this vector and all the other 512-dimensional vectors in the standard dataset is calculated as follows, where D_k represents the Euclidean metric between the 512-dimensional vector from the input and the *k*th music piece in the standard dataset:

$$D_k = \sqrt{\sum_{i=1}^{512}(x_{input}[i]-x_k[i])^2}$$

All the consecutive intervals of 64 128-dimension vectors were calculated respectively, and the smallest D_k among every consecutive interval and all 245 songs in the standard dataset is chosen. Also, the serial number of the smallest

D_k, k, is recorded. This means that the *k*th music piece in the standard dataset is the closest to the input.

Because the export vectors of VGGish [2] have integer values ranging from 0 to 255, the final rating of the piece is designed as follows:

$$R = \max\left(\left\lfloor \frac{256}{D_{min}} \times 100 \right\rfloor, 100\right)$$

The equation results in an integer score of R, ranging from 0 to 100. R is the ultimate output of the model, following the information of a music piece in the standard dataset that is closest to the input. This greatly improves the interpretability of the model.

6. Evaluation

Two evaluation methods are applied: an objective evaluation based on the standard deviation of the rating results, and a subjective evaluation considering the biases between model and manual rating.

6.1. Objective Evaluation

Standard deviation is usually applied in evaluating a rating model. It represents the degree of dispersion for a set of data and is defined as the following:

$$S = \sqrt{\frac{\sum_{i=1}^{n} (\bar{a} - a_i)^2}{n}}$$

The greater the dispersion of a dataset, the greater the standard deviation. Specifically, if all the data in one set is equivalent, the standard deviation of this dataset reaches the minimum of 0.

For multiple versions of a single music piece, the rating result of the model should be similar. Thus, a low standard deviation of a set with the different versions of the same music piece indicates a superior model performance. Also, non-musical audios should be rated significantly lower than music pieces, as they do not contain any musical features.

Table 1 shows the standard deviation of three sets of input data and their standard deviation results. The standard deviation remains relatively small, marking the reliability of the model.

Table 1. Comparison on Different Sets of Input Data.

Name	Sample Size	Mean Rating	In the Standard Dataset	Standard Deviation
Mendelssohn—Violin Concerto	10	69.0	True	5.138
Mozart—Divertimento K.136	7	72.7	False	4.061
Vocal Speeches (Non-Music)	8	36.0	False	6.325

6.2. Subjective Evaluation

Subsequently, the subjective evaluation method is applied. 30 random pieces of classical music are chosen and rated manually. Then, these manual ratings were compared with the rating results of the model.

The mean of the difference between the ratings outranges 15. However, it must be stated that this is a very subjective assessment and personal attitudes were likely brought in. Still, in this case, 12 out of 30 of these ratings have differences below 10. Also, after removing 7 elements with the largest differences, the mean difference reaches 9.91, indicating the model is close to human preferences.

7. Limitations and Improvements

Besides its significance, the model still has a few limitations. Some major drawbacks of this model and measures for improvement are listed below.

7.1. Problems

• Feature Extractor is Not Suited for Musical Analysis: Although the VGGish [2] model is powerful and performs well on general audio, it is not specified for musical analysis. Therefore, it is not sensitive to melody and harmony features in the audio. This model likely overlooked these melodic and harmonic features, resulting in significant differences between manual rating and model rating in a few cases.

• VGGish Extractions Inconsistent: The small 0.96s interval of VGGish extraction makes the results, the 64x128 matrix, fractured. Although they were compressed into a single vector, this compression is not ideal considering the overall structure of the audio. Macro and structural features were neglected.

7.2. Solutions

• As for the suitability of the feature extractor, a large dataset specified for classical music analysis could be constructed using the pure audio of classical music pieces and tagging their genre, form, composer, instruments, and tonality of a piece. Also, a reformed neural network seems applicable. The data-driven harmonic filters [6] seem to be a promising solution.

• As for the inconsistency of VGGish, it could be solved by training another model with a larger interval as input. In this way, the network could learn the characteristics of macroscopic structures and result in a higher accuracy.

8. Conclusions

This paper introduces a model designed to rate the "public acceptance" score of classical music pieces. Four modules were devised to perform resampling and normalization, feature extraction, comparison, and rating. It is proven by both objective and subjective evaluation that the results are trustworthy. After probable improvements and applications in the future, this model could serve as a much more reliable reference for further research in musical evaluation, musical selection, and musical recommendation.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data that support the findings of this study are available from the author, Y. Hong, upon reasonable request.

Conflicts of Interest

The author declares no conflict of interest.

References

1 236,000 Works Have Scores or Parts on Petrucci Music Library. Available online: https://imslporg (accessed on 31 August 2024).
2 VGGish. 2023. Available online: https://github.com/tensorflow/models/tree/master/research/audioset/vggish (accessed on 31 August 2024).
3 Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 2014; arXiv:1409.1556.
4 Acharya A, Das S, Pan I, *et al*. Extending the Concept of Analog Butterworth Filter for Fractional Order Systems. *Signal Processing* 2014; **94**: 409–420.

5 Classical Music Only. The Greatest 100 Composers of All Time. Available online: https://classicalmusiconly.com/lists/top/composers (accessed on 31 August 2024).

6 Won M, Chun S, Nieto, O, *et al*. Data-Driven Harmonic Filters for Audio Representation Learning. In Proceedings of the ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 536–540.