Predicting Audio Features of Background Music from Game Scenes

Ryusei Hayashi[1] and Tetsuro Kitahara[1⋆]

[1]Graduate School of Integrated Basic Sciences, Nihon University
Setagaya-ku, Tokyo, Japan
ryusei@kthrlab.jp

Abstract. *We propose a system to retrieve background music (BGM) for game scenes. BGM plays an important role in creating a particular atmosphere in game scenes, so studies have investigated the relationship between game scenes and BGM. However, none of the existing studies attempted to predict the audio features of BGM directly from a sequence of images expressing game scenes. In our system, the user inputs a sequence of images of a game scene, then our machine learning model, trained with gameplay videos, predicts the audio features from the input. Finally, the system retrieves the closest musical piece to the predicted audio features. Experimental results show both positive and negative tendencies: the predicted audio features for fight scenes are closer to the features of actually used BGM in fight scenes than those in other scenes (positive); the same musical piece was retrieved for different scenes (negative).*

Keywords: CNN-LSTM, Video Game Music (VGM), Role-playing Game (RPG), Speedrun Video, Copyright-free Music

1 Introduction

Background music (BGM) plays a role in creating the atmosphere of a video game. In particular, musical pieces used in different scenes (e.g., talk, fight) would be carefully composed to make different atmospheres that different scenes have. Therefore, we suppose a strong relationship exists between the atmosphere of a scene, and the feature of the BGM used there. For example, the BGM in a fight scene of a role-playing game (RPG) may tend to create a tense atmosphere with strong beats, while the BGM used in a talk scene may tend to create a calm atmosphere with soft timbres and rhythm.

The final goal of our study is to establish a technology that makes it possible to recommend the BGM that fits each of the various scenes in a game. This technology is intended to be used by indie game creators. After they create various scenes of their own game, they will give each scene (a sequence of screen images) to our system. Then, the

⋆ This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.

system recommends the musical piece that fits each scene as BGM based on a machine learning model.

Some researchers have investigated the relationship between game scenes and BGM. Yamauchi et al.[1] developed a system that retrieves music from game scenarios. Nemoto et al.[2] investigated the relationship between the emotional state of a character and BGM. Choi et al.[3] created a game music database with emotion labels and a model to generate a baseline. Kim et al.[4] developed a game that records BGM's and the player's moods simultaneously and analyzed their relationship. Ishikawa et al.[5] developed a system that retrieves BGM from visual scenes through impression words. Zeng et al.[6] developed a system that retrieves movies from music. "AmBeat"[7] is an application that adds generated music to a video when the video is input. "Deep12"[8] searches for similar music when music is input. They did not deal with directly predicting musical features of BGM for games from those games' visual scenes.

In this paper, we develop a method for predicting audio features of BGM that fits given game scenes from a sequence of screen images. Because there are many gameplay videos on Web-based video hosting services, we can quickly obtain large-scale data consisting of pairs of a sequence of screen images of a game and audio signals of its BGM. By learning those data, we will achieve the prediction of the audio features of BGM from screen images.

2 Proposed Method

We propose a system that outputs the audio features of BGM suitable for the input game scene. The system is intended to be used when the user creates his/her own game and finds musical pieces for adding to the game as BGM. First, the user inputs a sequence of images of a scene (e.g., Fight, Talk) included in the created game. Then, the system predicts a sequence of the audio features that are considered to fit the given scene. Finally, the system outputs the musical piece with the closest audio features to the predicted ones from the music collection prepared in advance.

It is generally challenging to find a universal relationship between scenes and BGM. We assume that the user creates a game referring to an existing game (called a *referred game* here), and they are similar to each other. Therefore, we let the user specify the referred game and learn the relationship between scenes and BGM.

2.1 Input and Output Data

The input and output data were taken from speedrun videos posted on YouTube. First, we saved videos in MP4 format. Next, we classify the video frame by frame using k-means[9]. Finally, we extracted one hundred 12-second segments from the video to avoid including multiple classes.

Then, we applied the following pre-processing. We divide the input and output data obtained by these processes in half and use them as training and test data.

Input Data We loaded a 12-second video using the OpenCV library and converted the color space of images from BGR to HSV. The image size was also changed to 80×80. We accordingly obtained tensor data of dimensions $80 \times 80 \times 3$.

Output Data We extracted the audio tracks from the videos mentioned above and saved them in WAV format. The audio features described in Table 1 were extracted using the LiBROSA library. Some of these audio features can be selected and used.

Table 1. Audio features to be extracted

	Feature	Outline
01	cqt	Semitone power specrogram using constant-Q transform
02	iirt	Semitone power spectrogram using a multirate filter bank consisting of IIR filters
03	chroma_stft	12-dimensional features representing the power of each pitch class, calculated from the STFT-based power spectrogram
04	chroma_cqt	12-dimensional features representing the power of each pitch class, calculated from the CQT-based power spectrogram
05	chroma_cens	12-dimensional features with smoothed temporal variations in chroma_cqt
06	melspectrogram	Mel-scaled spectrogram
07	mfcc	Mel-frequency cepstral coefficients
08	mfcc_delta2	Temporal second-order differentials of MFCCs
09	nmf	Activations obtained by non-negative matrix factorization from the spectrogram

2.2 Model Architecture

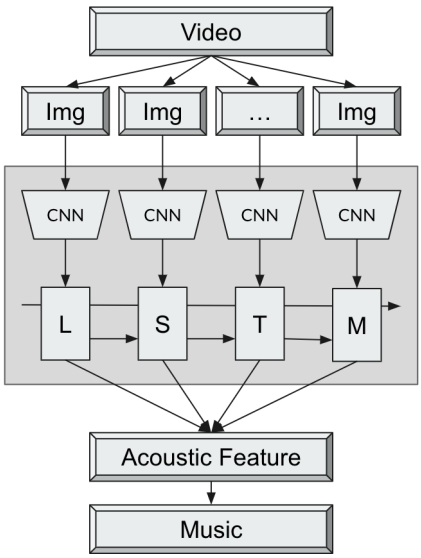
Our model is based on CNN-LSTM[10][11], in which the CNN[12] part reduces the image data of the given game scene video while the LSTM[13] part models the temporal features contained in the scene video and BGM. The overview of this model is shown in Fig. 1. The CNN part consists of multiple convolution layers and max pooling layers. The LSTM part consists of two LSTM layers to make it possible to consider long temporal dependencies.

As mentioned above, we assume that the user has a *referred game*, an existing game that he/she referred to when creating a game. Therefore, our model is trained individually on each training game, and the model trained on the referred game is intended to be selected by the user.

This model has been implemented with the Keras library of Tensorflow. We use ADAM[14][15] as an optimizer and the mean squared error as the loss function. The batch size is 16. The number of epochs is 500 for chromagrams and 100 for other audio features. They were experimentally determined to make the loss function less than 0.01.

2.3 Retrieval of musical pieces from predicted audio features

After the audio features for BGM are predicted, the system retrieves the musical piece with the closest audio features to the predicted ones from a pre-made music collection. This process includes the following two phases.



	Layer type	Details	Activation
01	Convolution	Filters: 16, Size: 3×3	ReLU
02	Max pooling	Size: 2×2	—
03	Convolution	Filters: 32, Size: 3×3	ReLU
04	Max pooling	Size: 2×2	—
05	Convolution	Filters: 64, Size: 3×3	ReLU
06	Max pooling	Size: 2×2	—
07	Flatten	—	—
08	Full connect	Nodes: 128	ReLU
09	Full connect	Nodes: 128	—

Fig. 1. Architecture of our model. The right-hand table shows the details of the CNN layers

Extraction of 12-second representative segment First, the system extracts a 12-second representative segment from each piece included in the collection. This is because a sequence of audio features extracted from each piece in the collection should have the same duration as the predicted audio features. Although extracting the first 12-second segment may be the simplest way, it may not capture the characteristics of the entire music. Therefore, we extract the segment that captures the characteristics of the music as follows:

1. A sequence of MFCCs is extracted from the target audio signal.
   Let $\mathbf{x} = [x_1, x_2, \cdots, x_N]$ be the sequence of the MFCCs.
2. All MFCC vectors $x_1, x_2, \cdots, x_N$ are clustered with the k-means algorithm[16]. The number of clusters is set to 4. Let c_i be the cluster ID of x_i. Then, the most frequent cluster ID, c_{mode}, is obtained.
3. Let $\mathbf{x}_i = [x_i, x_{i+1}, \cdots, x_{i+n}]$ be a 12-second segment beginning at x_i, where n is the number of elements for a 12-second segment. Then we compute $\hat{i}$ that satisfies the following equation:

$$\hat{i} = \underset{i \in [0, N-n]}{\operatorname{argmax}} \ \mathrm{count}(c_{\mathrm{mode}}, [c_i, c_{i+1}, \cdots, c_{i+n}]),$$

   where $\mathrm{count}(a, A)$ counts how many elements in a sequence A equals a.
4. $\mathbf{x}_{\hat{i}}$ is regarded as the 12-second representative segment.

Search of musical piece Next, we extract audio features from the extracted 12-second representative segment for every piece in the collection. The audio features to be ex-

tracted, listed in Table 1, are the same as those used in the prediction with the CNN-LSTM model. Then, the Earth Mover's Distance[17][18] of the extracted audio features from the predicted ones is extracted for every piece in the collection. Finally, the piece that has the minimal distance is searched.

3 Experiments

We conducted the following experiments.

1. Determination of referred and test games
2. Prediction of audio features
3. Retrieval of musical pieces from the predicted audio features

3.1 Dataset

We made a dataset from speedrun videos of the games in Table 2 posted on YouTube. We divided them into 12-second scenes and extracted two fight scenes, two walk scenes, and two talk scenes. We created a music collection for BGM from the copyright-free music sites in Table 3. We downloaded 99 WAV files from free music sites.

3.2 Determination of referred and test games

As mentioned above, we assume that the user selects a referred game and uses the model trained with that game. To simulate this situation, we adopt the following three-step approach. To reduce the computation time for learning models, we first choose a referred game and then decide the test game which is closest to the referred game.

1. Choose a referred game.
2. Find the game with the most similar visual scenes to the chosen one. This game is regarded as a test game.
3. The model trained with the referred game's data is used for predicting audio features.

Step 2 is calculated based on the average of the image hash value differences. The average hash value difference is calculated in the following steps.

2-1 Load the videos of the two games as images.
2-2 Compute hash values of all images.
2-3 Compute the difference between the hash values of the two games for all combinations.
2-4 Compute the average of hash value differences.

The results of the calculated dissimilarities are shown in Fig. 2. For "Undertale" (referred game), "OMORI" (test game) was selected. For "Chrono Trigger" (referred game), "Romancing Saga 3" (test game) was selected. Fig. 3 shows some excerpts of the visual scenes of those games.

Table 2. Games used in the experiment as referred and test games

	Game	Usage	Outline
01	Ghost of Tsushima	Test game	A samurai joins a battle on a quest to protect Tsushima Island during the first Mongol invasion of Japan
02	OFF	Test game	An enigmatic humanoid entity called *Batter* travels the world on a *sacred mission* to *purify* the world
03	OMORI	Test game	The player explores both the real world with Sunny and the dream world with his alter-ego "OMORI" in the dream, overcoming his secrets
04	Undertale	Referred game	The player controls a child who has fallen underground and adventures back to the surface while meeting various monsters
05	Chrono Trigger	Referred game	The player controls a group of adventurers on a journey through time to prevent a global catastrophe
06	It's a Wonderful World	Test game	Players are deprived of what is most precious to them and forced to participate in the Reaper's Game for the survival of Shibuya
07	NieR:Automata	Test game	Players take on the role of human-made androids in a proxy war against an invading army of Machines from another world
08	PERSONA5	Test game	The player and his friends awaken their persona abilities and become the Phantom Thieves of Hearts to steal malevolent intent from the hearts of adults
09	MONSTER HUNTER STORIES	Test game	Players explore the world after the village where they live with the monsters they were born into is hit by a disaster
10	Romancing Saga 3	Test game	Rise of Morastrum occurs again, and the player ends up involved in the hunt for the Child of Destiny as eight main characters
11	WILD ARMS	Test game	Players control a boy who wields ARMS to prevent an otherworldly threat from reviving their lost leader and destroying the world
12	Okami	Test game	The player becomes Amaterasu and embarks on a journey to fulfill people's wishes to defeat Yamata no Orochi and restore the world

Table 3. Copyright-free music websites used to create a music collection

	Site	URL
01	bensound	https://www.bensound.com/
02	DOVA-SYNDROME	https://dova-s.jp/
03	MusMus	https://musmus.main.jp/
04	PeriTune	https://peritune.com/
05	Solitary Sound	https://az-ho.org/
06	Devil Soul	https://maou.audio/

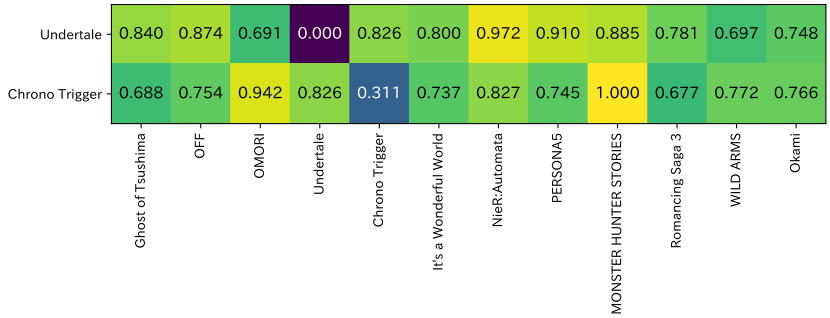


Fig. 2. Dismilarity of visual scenes between games to determine test games

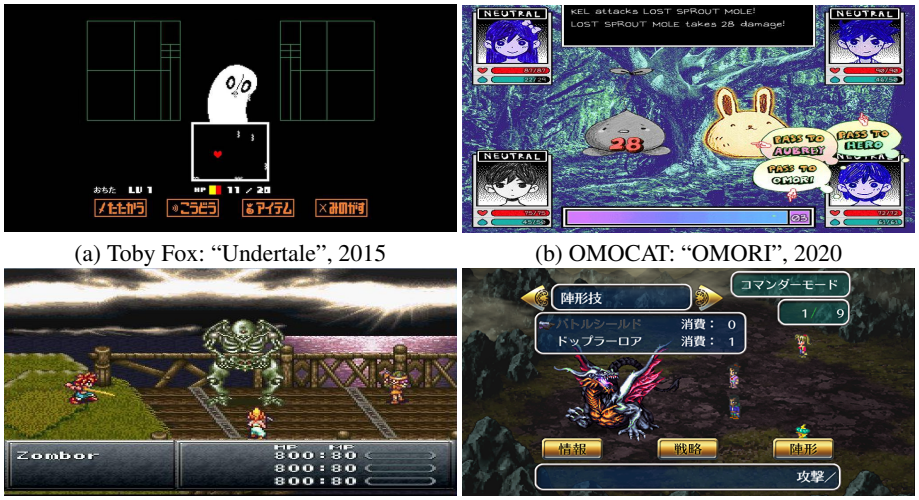
3.3 Evaluation on prediction of audio features

We experimented with evaluating audio feature prediction through our CNN-LSTM model. This evaluation compares the dissimilarity between the predicted audio features and those of actually used BGM. Because the effectiveness of the prediction would be different among audio feature categories, the effects of each feature category are evaluated individually, as well as their combinations.

Method We trained models with each feature category to evaluate the effects of each audio feature category individually. Because we have nine feature categories (Table 1) and two games ("Undertale" and "Chrono Trigger") as referred games, we trained 18 models. Then, we gave the models six scenes S (two fight, two walk, and two talk scenes) from each of the two games ("OMORI" and "Romancing Saga 3") as test games. Hence, we obtained the predicted audio features $y_i^{\mathrm{pred}}(s)$ and compared them with the audio features $y_i^{\mathrm{true}}(s)$ of actually used BGM (i: audio feature type, $s(\in S)$: scene).

Because BGM for different scenes should have different audio features, we identified that the audio features have inter-scene variations. Specifically, we calculate i that maximizes the following equation:

$$\sum_{s\in S}\sum_{s'\in S\setminus\{s\}}\{\mathrm{dist}(y_i^{\mathrm{true}}(s'),y_i^{\mathrm{pred}}(s))-\mathrm{dist}(y_i^{\mathrm{true}}(s),y_i^{\mathrm{pred}}(s)\}$$



(a) Toby Fox: “Undertale”, 2015 (b) OMOCAT: “OMORI”, 2020

(c) SQUARE: “Chrono Trigger”, 1995 (d) SQUARE: “Romancing Saga 3”, 1995

Fig. 3. Excerpts of visual scenes in the used games

where $\mathrm{dist}(y_i^{\mathrm{true}}(s), y_i^{\mathrm{pred}}s)$ represents the distance between the predicted and actual audio features of the same scene.

Results For “OMORI” (referred game: “Undertale”), the use of only chroma_cqt maximized the inter-scene variations of the audio features. For “Chrono Trigger’ (referred game: “Romancing Saga 3”), the combination of chroma_stft, chroma_cqt, chroma_cens, and mfcc_delta2 maximized the inter-scene variations.

Discussion Fig. 4 (Left) shows the distance between the predicted features and actual features of each scene for “OMORI” (referred game: “Undertale”). Observations from this figure can be summarized as follows:

- When we focus on Fight Scene 1’s predicted features, the distance from the actual features of the same scene should have been the smallest, but the distance from Walk Scene 1’s actual features was the smallest. Also, for Fight Scene 2, the distance of its predicted features from Walk Scene 1’s actual features was the smallest. These results imply that “Undertale” fight scenes and “OMORI” walk scenes may have similar features in BGM.
- When we focus on the two walk scenes, both Walk Scene 1’s and Walk Scene 2’s predicted features had the smallest distance from Walk Scene 1’s actual features. It means that our models well predicted the walk scenes’ audio features.
- For the two talk scenes, the distances between their predicted and actual features were the largest. In general, talk scenes’ actual features tended to have large distances from all scenes’ predicted features. This could be why “Undertale” tended to have few talk scenes.

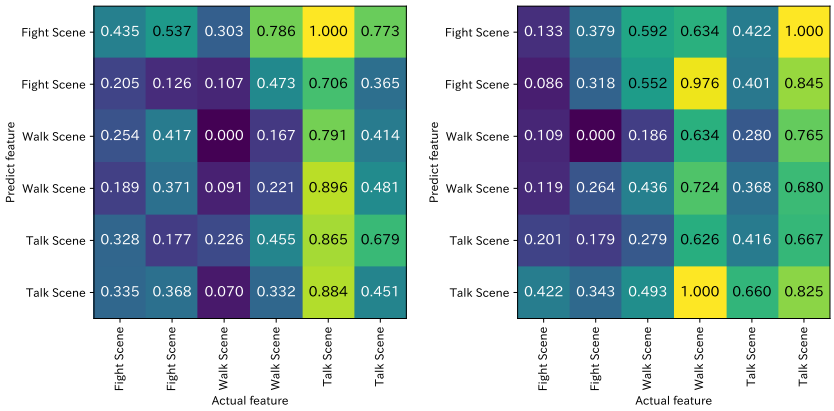


Fig. 4. Distance matrix of predicted vs. actual audio features (Left: "Undertale", Right: "Chrono Trigger")

Fig. 4 (Right) shows the distance between the predicted features and actual features of each scene for "Chrono Trigger" (referred game: "Romancing Saga 3"). Observations from this figure are summarized as follows:

- When we focus on the two fight scenes, the scenes with the actual features with the minimal distance from the fight scenes' predicted features were fight scenes. These results imply that "Chrono Trigger"'s fight scenes may have similar features BGM to each other.
- The walk scenes' predicted features were closer to the fight scenes' actual features than the walk scenes' ones. Similarly, the talk scenes' predicted features were also closer to the fight scenes' actual features than the talk scenes' actual features. As well as "OMORI", the distance from the talk scenes' actual features tended to be large in general. This could be why "Chrono Trigger" tended to have many fight scenes.

3.4 Evaluation on retrieval of musical pieces from the predicted audio features

We experiment with music output. If the same music is output, even if different scenes are input, it goes against the purpose of the research. Therefore, we verify which audio features are suitable for outputting different music.

Method We prepared six scenes from each of the 12 games listed in Table 2 (72 scenes in total). Let $S = \{s_1, s_2, \cdots, s_J\}$ be a set of the prepared scenes. For BGM retrieval, we used the music collection described in Section 3.1, which consists of 99 musical pieces taken from copyright-free music collection websites. Here, $M = \{m_1, m_2, \cdots, m_K\}$ be the music collection. For each scene s_i in S, we retrieved the musical piece that best fits the given scene. Here, the retrieved musical piece for scene s_i is represented by $\mathrm{output}(s_j)$.

The important point is that retrieved musical pieces should differ for different scenes. In other words, for s_i and s_j ($i \neq j$), it should be $\mathrm{output}(s_i) \neq \mathrm{output}(s_j)$. Therefore, for each musical piece m_k, we calculated the number of scenes, $s_j (1 \leq j \leq J)$ satisfying $m_k = \mathrm{output}(s_j)$. This number is denoted by $X_i(m_k)$ (i: the audio feature category). When the condition mentioned above is satisfied, $X_i(m_k)$ should equal 0 or 1 (as long as M has a sufficiently large number of pieces compared to the number of scenes), and its expected value is J/K. Therefore, we evaluated the appropriateness of the BGM retrieval by calculating the mean squared error between $X_i(m_k)$ and J/K. That is, we identified the most effective audio feature category $\hat{i}$ that minimizes the following equation:

$$\hat{i} = \underset{i \in I}{\mathrm{argmin}} \sum_{k=1}^{K} \left(X_i(m_k) - \frac{J}{K} \right)^2$$

Results Experimental results show that chroma_cqt is the most effective audio feature category for learning "Undertale". Table 4 lists the retrieval results obtained by inputting the scenes of "OMORI" into the model trained by "Undertale" with chroma_cqt. This shows that the same musical piece was output for different scenes (the two walk scenes and one talk scene).

For "Chrono Trigger", chorma_stft is the most effective audio feature category. Table 5 lists the retrieval results obtained by inputting the scenes of "Romancing Saga 3" into the model trained by "Chrono Trigger" with chroma_stft. It shows that the same musical piece was output for Walk Scene 1 and Talk Scene 2. Otherwise, different musical pieces were output for different scenes.

Table 4. Musical pieces retrieved for scenes from "OMORI" (features: chroma_cqt, referred game: "Undertale")

	Scene	EMD	Music title	Artist	URL
01	Fight Scene 1	0.132	Catch!!	watson	https://musmus.main.jp/music_game.html
02	Fight Scene 2	0.075	And then we ran	watson	https://musmus.main.jp/music_game.html
03	Walk Scene 1	0.044	Pursuer	watson	https://musmus.main.jp/music_game.html
04	Walk Scene 2	0.024	Pursuer	watson	https://musmus.main.jp/music_game.html
05	Talk Scene 1	0.125	And then we ran	watson	https://musmus.main.jp/music_game.html
06	Talk Scene 2	0.051	Pursuer	watson	https://musmus.main.jp/music_game.html

Table 5. Musical pieces retrieved for scenes from "Chrono Trigger" (features: chroma_stft, referred game: "Romancing Saga 3")

	Scene	EMD	Music title	Artist	URL
01	Fight Scene 1	0.078	And then we ran	watson	https://musmus.main.jp/music_game.html
02	Fight Scene 2	0.067	Pursuer	watson	https://musmus.main.jp/music_game.html
03	Walk Scene 1	0.033	Sonorously Box	watson	https://musmus.main.jp/music_game.html
04	Walk Scene 2	0.036	The Chuckling Witch	Hibiki Abe	https://az-ho.org/a-smiling-witch
05	Talk Scene 1	0.088	Mid-range Strength	watson	https://musmus.main.jp/music_game.html
06	Talk Scene 2	0.086	Sonorously Box	watson	https://musmus.main.jp/music_game.html

4 Conclusion

In this paper, we proposed a system that retrieves BGM that fits game scenes given as a sequence of screen images. Using gameplay videos taken from YouTube, we learned a CNN-LSTM-based transformation model from a sequence of screen images to audio features of BGM. Next, the system uses this model to predict the audio features that match the given game scene as BGM. Finally, the system retrieves the musical piece with the closest audio features to the predicted ones.

To confirm the effectiveness of this system, we conducted some experiments. In particular, the comparisons of the predicted audio features and those used in actual BGM show that the predicted features for fight scenes are close to those of the actual BGM. In contrast, the predicted features of walk scenes and talk scenes are not close to those of the same scenes' actual BGM. Also, we discussed retrieved musical pieces for each scene. Retrieved musical pieces should be different for different scenes. It was partly achieved, even though the same musical piece was output for some scenes.

This research is based on a strong assumption that screen images and BGM in games have explicit dependencies on each other. We believe this assumption is partly true but has not yet been fully confirmed. In the future, we will verify the appropriateness of our ideas with larger-scale data as well as the system's usability tests.

References

1. T. Yamauchi, S. Nemoto, K. Nagano, S. Nakamura, A. Uda, Y. Saito, H. Murai, E. Tayanagi, K. Mukaiyama, and K. Hirata: "Game BGM Selection Based on Scenario and Emotional State", The 34th Annual Conference of the Japanese Society for Artificial Intelligence, pp.1–3, 2020 in Japan.
2. S. Nemoto, K. Ishikawa, A. Uda, T. Shiraishi, S. Nakamura, K. Nagano, T. Yamauchi, H. Murai, K. Hirata, K. Mukaiyama, and E. Tayanagi: "Extraction of Relationship between Character's Emotional State and BGM in Story Scene", JSIK, Vol.30, No.2, pp263–269, 2020 in Japan.
3. E. Choi, Y. Chung, S. Lee, J. Jeon, T. Kwon, and J. Nam: "YM2413-MDB: A Multi-Instrumental FM Video Game Music Dataset with Emotion Annotations", ISMIR, arXiv:2211.07131, 2022.
4. Y. E. Kim, E. Schmidt, and L. Emelle: "Moodswings: A collaborative game for music mood label collection", ISMIR, pp.231–236, 2008.
5. T. Ishikawa: "Background Music Search System to an Input Video Using Factor Analysis for Impression Words", IIEEJ, Vol.9, No.2, pp.69–77, 2023 in Japan.
6. D. Zeng, Y. Yu, and K. Oyama: "Audio-Visual Embedding for Cross-Modal Music Video Retrieval through Supervised Deep CCA", IEEE ISM, pp.143–150, 2018.
7. AmBeat, `https://tollite.yamaha.com/Ambeat/`
8. Deep12, `https://www.sonycsl.co.jp/tokyo/14621/`
9. J. MacQueen: "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.1, pp.281–297, 1967.
10. N. Somu, R. Gauthama, and R. Krithivasan: "A deep learning framework for building energy consumption forecast", Renewable and Sustainable Energy Reviews, Vol.137, 2021.

11. R. Rial, A. R. R. Adhitya, and L. Hyun-Jin: “A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power”, Energies, Vol.13, 2019.
12. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner: “Gradient-based learning applied to document recognition”, IEEE, Vol.86, No.11, pp.2278–2324, 1998.
13. S. Hochreiter and J. Schmidhuber: “Short-Term Memory”, Neural Computation, Vol.9, No.8, pp.1735–1780, 1997.
14. D. P. Kingma and J. Ba: “Adam: A Method for Stochastic Optimization”, ICLR, arXiv:1412.6980, 2015.
15. S. J. Reddi, S. Kale, and S. Kumar: “On the Convergence of Adam and Beyond”, ICLR, arXiv:1904.09237, 2018.
16. B. McFee and Daniel P. W. Ellis: “Analyzing Song Structure with Spectral Clustering”, ISMIR, 15th International Society for Music Information Retrieval Conference, pp.405–410, 2014.
17. B. Logan and A. Salomon: “A music similarity function based on signal analysis”, IEEE ICME, pp.745–748, 2001.
18. Q. Xiao, S. Tsuge, and K. Kita: “Music retrieval method based on filter-bank feature and earth mover’s distance”, Seventh International Conference on Natural Computation, pp.1845–1849, 2021.