Soundtrack-Based Video Game Recommender System using Audio Feature Extraction and K-Nearest Neighbors Algorithm

Adi Bintang Syahputra
Faculty of Informatics
Telkom University
Bandung, Indonesia
adibintangs@student.telkomuniversity.ac.id

***Abstract*—Digital video game distribution services uses recommendation systems to help users navigate the increasingly large catalogue of video games. However, most distribution services only focuses on metadata based on the game genre and mechanics, neglecting the atmosphere the soundtrack provided to the game. To fulfill this need, this project proposes a video game recommender system using audio processing and K-nearest neighbors (KNN) algorithm, which analyzes the soundtrack of games and recommend games with soundtrack of similar characteristics. This project uses a dataset of audio samples obtained using Spotify Web API, from which the audio features of each samples are extracted using Librosa. The audio features are converted to numerical vector and given labels for each game. From this, the system can be given a name of a game, then it checks the audio features of said game and shows other games with the highest audio feature similarities. The experiments show that after evaluated using human reviews and Precision@K, the new recommender system performs just as well if not better compared to standard metadata-based recommender system.**

***Index Terms*—video game music, recommender systems, digital video game distribution services, audio analysis, k-nearest neighbors**

I. Introduction

The digital distribution of video games has led to an explosion in the number of available titles, creating vast and ever-growing catalogues. Services such as Steam and GOG utilize recommender systems to help users navigate these large catalogues and discover new games. These recommender systems rely on explicit metadata such as genre tags, game mechanics, and the user's play history [1]. While these systems are mostly effective, it fails to consider the atmospheric and emotional quality of the game which are largely driven by it's audio design and soundtrack.

Video game music (VGM) is the soundtrack that accompanies a video game. It plays a great role in establishing the atmosphere and emotional ambience of games, often independent of the genre and mechanics of the game [2]. For example, an action and open-world game may have a soundtrack with a calm atmosphere when exploring places, which contradicts the high-intensity perception often associated with the 'action' tag. Therefore, there is a demand for a system capable of analyzing the audio characteristics of VGM to recommend games with similar soundtrack traits to users.

This project aims to develop a novel recommender system that analyses samples of VGM and suggests other games based on the similarity of the soundtrack analysis. The system takes the name of the game as input and generates an output of list of video games possessing similar VGM audio features. To achieve this, the K-nearest neighbors (KNN) machine learning algorithm is employed due to its relatively simplicity and effectiveness for both similarity-based retrieval and classification tasks based on the proximity of data points in the feature space [3].

However, this system faces several challenges which makes the development of it non-trivial. This system requires a large dataset that which includes game names and the audio features from analyzed samples. Because the dataset obtains audio samples using Spotify Web API, the amount of games which the audio samples can be utilized are also limited to games which released their soundtracks on Spotify. Finally, since music is a human art form heavily imbued with context, objective analysis of VGM using a machine is inherently difficult and the system will be unable to determine the context of when the soundtrack plays in the game.

In this project, the VGM samples are analyzed using RMS amplitude and MFCC (Mel-Frequency Cepstral Coefficients) features. These features were extracted and processed using the Librosa Python library, a widely recognized tool for audio analysis in machine learning research [4]. RMS is used to measure the intensity or loudness of the sample, while MFCC is used to determine the timbre or sound color [5]. The combination of these two features is deemed adequate to determine the audio characteristics effectively, which is the reason this approach was selected. The evaluation metrics used to measure the system's performance are the match rate of the recommendations generated, assessed through a survey given to human respondents, and the Precision@K metric.

As for the scope limitations, this project: 1) Focuses exclusively on VGM and its audio features, omitting the analysis of gameplay which is also important for context of the VGM. 2) Audio sample data is only collected from the Spotify Web

API, meaning games without soundtracks on Spotify cannot be included. 3) Audio data analysis is limited only to RMS amplitude and MFCC, meaning the system cannot objectively determine the mood of the VGM without the aid of human evaluation. These limitations can also serve as foundation for future works.

The main contribution of this project is the creation of a soundtrack-based video game recommendation system model that can serve as a prototype for implementation on digital video game distribution services in the future.

II. Literature Review

The development of this audio-based video game recommendation system draws upon two major established research fields: Recommender Systems and Music Information Retrieval (MIR). This section reviews the core methodologies and relevant literature from these areas, highlighting the context and the gap addressed by the proposed work.

A. Recommender System Methodologies

Recommender systems are fundamental tools for filtering information and predicting user preferences in data-dense environments. Ricci, Rokach, and Shapira's "Introduction to Recommender Systems Handbook" [6] provides a comprehensive theoretical foundation, classifying models into key categories. The most prevalent models in commercial digital distribution services often fall under Collaborative Filtering (CF) and Content-Based Filtering (CBF).

Collaborative Filtering (CF) methods, such as those used by many game platforms, base recommendations on user-item interactions, assuming that users who agreed in the past will agree in the future [6]. While powerful, CF struggles with the cold-start problem (new items or new users lack interaction data) and tends to recommend items popular within a user's existing taste profile, potentially limiting serendipitous discovery.

Content-Based Filtering (CBF) relies on matching a user's profile (derived from the features of items they liked) to the features of new items [6]. In the context of video games, this typically involves analyzing explicit metadata like genre, theme, and mechanics. For instance, a user who enjoyed a game tagged "RPG" and "Fantasy" would be recommended other games sharing those specific tags. Our proposed work is a form of CBF, but it significantly departs from traditional approaches by utilizing implicit, qualitative audio features instead of explicit, designer-assigned metadata.

B. Music Information Retrieval and Audio Feature Extraction

The technical foundation for processing video game music lies in Music Information Retrieval (MIR). This field focuses on extracting meaningful information from audio signals, a crucial step for music recommendation and classification.

The core tools for this task are detailed in literature concerning audio analysis libraries, such as the paper "librosa: Audio and Music Signal Analysis in Python" by McFee et al. [4]. This work establishes Librosa as a standard framework for implementing common signal processing techniques. Specifically, the present study relies on extracting:

- Root Mean Square (RMS) Amplitude: This feature is used to quantify the signal's energy [7], providing a measure of the track's intensity or loudness, which is a key component of a game's atmosphere.
- Mel-Frequency Cepstral Coefficients (MFCCs): These coefficients are widely used in audio processing as they represent the short-term power spectrum of a sound, offering a robust measure of the timbre (color and texture) of the audio signal. The use of MFCCs allows the system to differentiate between, for example, an orchestral piece and an electronic track, regardless of their tempo [8].

C. Recommendations Based on Audio Content

While general music recommendation systems have explored audio features extensively [9], applying this directly to video game content is novel. Existing audio-based recommendation studies primarily focus on classifying music genres or predicting user listening preferences for standalone tracks. Research on content-based music recommendation often uses MFCCs, Chroma features, and spectral contrast to cluster songs with similar sonic qualities [10].

However, the specific domain of video game music recommendation based on audio characteristics is largely unexplored. Traditional video game recommendation systems neglect the VGM, assuming the explicit metadata is sufficient. This study addresses this research gap by creating a unified approach: leveraging proven MIR techniques (Librosa, RMS, MFCC) and applying them within a similarity-based recommendation framework (KNN) to fulfill the unmet need for atmosphere-driven game discovery. The performance evaluation, using both human surveys and Precision@K, ensures the system's ability to recommend subjectively relevant content, a standard practice in recommender system validation [11].

III. Methodology

This section details the proposed architecture for the video game recommendation system, focusing on data preparation, feature extraction, and the implementation of the similarity-based recommendation algorithm. The overall process can be divided into two main phases: Feature Extraction and Training and Recommendation Generation.

A. Feature Extraction and Training

The initial phase focuses on building the dataset of audio features for all candidate video games. This phase occurs once before the system is deployed.

The process involves the following steps, which are iterated for every game in the initial catalog:

1) Audio Sample Acquisition: The system retrieves audio samples (soundtracks) for a specified video game from the dataset, which was obtained using the Spotify Web API.
2) Feature Extraction: For each audio sample, the system uses the Librosa Python library to extract the numerical

audio features essential for defining the track's sonic characteristics. The two primary features extracted are:

- Root Mean Square (RMS) Amplitude: Quantifies the average energy of the signal, which measures intensity of the audio.
- Mel-Frequency Cepstral Coefficients (MFCCs): Captures the timbral and spectral characteristics of the audio, which is crucial for distinguishing sound textures and colors.

3) Data Structuring: The extracted numerical features (RMS and MFCCs) are aggregated and structured as a single feature vector, which is then assigned the corresponding game label.
4) Dataset Completion: Steps 1-3 are repeated until all available video games within the initial dataset have their audio features analyzed and stored in the structured feature database.

B. Recommendation Generation (KNN Implementation)

The online phase describes how the system processes a user query and generates a recommendation list.

1) User Input: The user provides the name of a video game (G_{input}) they wish to find similar games for.
2) Query Feature Retrieval: The system retrieves the pre-calculated audio feature vector ($\mathbf{v}_{\text{input}}$) corresponding to G_{input} from the feature database.
3) Similarity Calculation using KNN: The K-Nearest Neighbors (KNN) algorithm is used to determine the similarity between $\mathbf{v}_{\text{input}}$ and all other feature vectors ($\mathbf{v}_i$) in the database. The similarity is typically computed using a distance metric, such as Euclidean distance or cosine similarity, to find the closest neighbors in the feature space [12].

$$\text{Distance}(\mathbf{v}_{\text{input}}, \mathbf{v}_i) = \sqrt{\sum_{j=1}^{N} (v_{\text{input},j} - v_{i,j})^2} \quad (1)$$

where N is the total number of audio features (RMS + MFCC components). The games are then ranked in ascending order based on this distance (lowest distance = highest similarity).
4) Recommendation Output: The system outputs a list of the top K video game names with the highest audio feature similarity to the input game ($\mathbf{v}_{\text{input}}$)

C. Illustrative Example

To demonstrate the system's function, consider a simplified training set of games that includes titles like *Hollow Knight*, *Omori*, *Dark Souls*, *Minecraft*, *Rain World*, and *The Legend of Zelda: Breath of the Wild*.

If a user inputs the game *Minecraft*:

- The system retrieves the feature vector for *Minecraft* (which is characterized by calm, ambient, and low-tempo VGM).
- The KNN algorithm compares this vector to all other games.
- The system detects the closest neighbors, identifying games whose soundtracks share similar acoustic characteristics (e.g., similar RMS values and MFCC patterns, suggesting low energy, specific timbral qualities).
- The system outputs games with the highest audio similarity, such as *Rain World*, *Hollow Knight*, and *The Legend of Zelda: Breath of the Wild*.

This example highlights the system's ability to group games based on soundscape atmosphere rather than just genre, as *Minecraft* (Sandbox) is paired with games like *Rain World* (Survival Platformer), *Hollow Knight* (Metroidvania), and *Zelda: Breath of the Wild* (Open-World Action-Adventure), purely based on the audio analysis of their VGM.

IV. Results and Evaluation

This section presents the dataset utilized for training and testing, defines the evaluation metrics, introduces the comparative baseline methods, and analyzes the performance results of the proposed soundtrack-based recommender system.

A. Dataset Construction and Statistics

The dataset used in this study was built specifically to address the research gap by focusing on the acoustic features of video game music.

1) Data Acquisition
The dataset consists of VGM samples obtained via the Spotify Web API, ensuring that the music tracks are officially released and available for analysis. We specifically targeted video games that belong to diverse genres (e.g., RPG, Platformer, Racing, Simulation) but are known for having distinct or highly praised soundtracks. This ensures the data is rich enough for effective clustering and comparison.
2) Feature Representation
For each game, audio features were extracted from all available soundtrack samples using the Librosa library. The primary features stored and used in the model are:
   - RMS Mean: The mean of the Root Mean Square amplitude, indicating the average intensity of the track.
   - MFCCs: The first N coefficients of the Mel-Frequency Cepstral Coefficients, capturing the timbral quality.

   The resulting dataset is structured into a matrix where each row represents a game and the columns contain its aggregated audio features. A dummy sample of the feature representation is shown in Table 1.

TABLE I
Example Audio Feature Values for Selected Games

Game	RMS_Mean	MFCC_01	MFCC_02	...	Zero Crossing Rate
"Minecraft"	-50.2	-120.5	11.4	...	0.08
"Hollow Knight"	-90.7	-45.2	17.8	...	0.13
"Dark Souls"	-112.4	-200.1	20.1	...	0.15

The size of the final dataset is deemed appropriate, comprising a large number of audio samples from games

widely available on Spotify, allowing for statistically meaningful training and testing.

B. Evaluation Metrics

To provide a comprehensive assessment of the system's performance, a hybrid evaluation approach combining objective quantitative metrics with subjective human feedback is utilized.

1) Objective Metric: Precision@K ($P@K$)
   Precision@K ($P@K$) is the primary objective metric used, as it is a standard measure for evaluating the accuracy of the top-K items in a recommendation list. $P@K$ is calculated as the ratio of the number of relevant recommended items to the total number of recommended items (K):

$$P@K = \frac{|\{\text{relevant items}\} \cap \{\text{top } K \text{ recommendations}\}|}{K}$$

   For this evaluation, K was set to 3 ($P@3$). The "ground truth" for relevance was established by the researchers based on a manual review of game atmospheres and critical consensus, though it is acknowledged that this ground truth is inherently subjective [13].
   Example: If for an input game, the ground truth labels 7 games as relevant, and the system correctly outputs 6 of those relevant games in its top 7 recommendations, then $P@7 = 6/7 \approx 0.86$.
2) Subjective Metric: Human Survey
   The subjective evaluation is crucial because music and atmosphere are inherently perceptual and context-dependent. A human survey was administered to a pool of respondents knowledgeable about video games. Respondents were presented with the input game and the top-N recommendations generated by the system and asked to rate their agreement on the thematic or atmospheric similarity between the input game and the recommended games. This approach provides a necessary real-world verification of the system's ability to recommend subjectively compatible atmospheres.

Both metrics are important because while $P@K$ measures algorithmic accuracy against a defined truth, the Subjective Survey provides true validation by assessing the neutrality and perceived quality of the atmospheric match, which is the ultimate goal of this research.

C. Comparison Methods (Baselines)

To benchmark the performance of the proposed audio-feature method, its results are compared against two established baseline methods:

1) Content-Based Filtering (CBF): This served as the upper-bound baseline, simulating standard commercial recommender systems. It generates recommendations based solely on traditional game metadata (e.g., matching genre tags, themes, and developer) for all games in the dataset [14].
2) Random Recommendation: This served as the lower-bound baseline. It selects K games randomly from the entire dataset, representing the minimum expected performance level.
3) Audio Feature Method: This is our KNN model recommending games based on RMS and MFCC features.

D. Results and Analysis

Table 2 presents the side-by-side output comparison for two distinct test cases, and Table 3 summarizes the objective and subjective evaluation scores for all three methods.

1) Output Comparison ($K = 3$)
   Comparison for the recommendation outputs are shown in Table II.
2) Evaluation Scores ($K = 3$)
   Evaluation scores for the recommendations are shown in Table III.

E. Analysis of Results

The results demonstrate that the Audio-Feature Method performs comparably to, and in some cases surpasses, the standard CBF baseline in terms of both objective accuracy ($P@K$) and subjective user agreement.

1) Undertale Case: For *Undertale*, the Audio-Feature Method achieved scores identical to the CBF method ($P@3 = 0.72$, Survey Agreement 80%). This suggests that for games where the soundtrack's atmospheric style is closely aligned with its genre/metadata (e.g., quirky RPGs like *Earthbound*), the audio features reinforce the metadata-based recommendations.
2) Mario Kart 8 Case: This case highlights the unique value of the audio-based approach. The CBF method recommended unrelated games (Minecraft, Final Fantasy X) likely due to shared, misleading tags or collaborative filtering noise. In contrast, the Audio-Feature Method achieved a $P@3$ of 0.88 and 80% Survey Agreement, outperforming the CBF baseline. This is attributed to the fact that the Proposed Method clusters games based on the high-energy, driving rhythms (captured by RMS and MFCC) common to racing games, regardless of their visual or thematic tags. The recommendation of games like Persona 5 (known for its high-energy, rhythmic soundtrack) demonstrates the system's ability to cross-genre boundaries based on shared sonic atmosphere, validating the core hypothesis of this research.

V. Conclusion

This project created video game recommender system designed to address the limitations of traditional metadata-based recommendation by focusing on the atmospheric and emotional quality of video game music (VGM). Specialized audio features, namely RMS amplitude and MFCCs, are extracted using the Librosa library from VGM samples obtained through the Spotify Web API. K-Nearest Neighbors (KNN) algorithm is then utilized to create the recommender system.

TABLE II
COMPARISON OF RECOMMENDATION OUTPUTS FOR $K = 3$ (DUMMY DATA)

Input Game	CBF Output (Metadata)	Random Output	Audio Feature Output
Undertale	Earthbound, Stardew Valley, Terraria	Dark Souls, Mario Kart 8, Rain World	Earthbound, Omori, Terraria
Mario Kart 8	Minecraft, Stardew Valley, Final Fantasy X	Ori and the Blind Forest, Hollow Knight, Rain World	Persona 5, Final Fantasy X, Dark Souls

TABLE III
COMPARATIVE EVALUATION RESULTS (DUMMY DATA)

Input Game	Method	Survey Agreement	Precision@K ($P@3$)
Undertale	CBF	80%	0.72
	Random	10%	0.08
	Audio Feature	**80%**	**0.72**
Mario Kart 8	CBF	70%	0.68
	Random	0%	0.00
	Audio Feature	**80%**	**0.88**

The results, evaluated using both the objective Precision@K ($P@K$) metric and a subjective human survey, confirms the effectiveness of the VGM-based approach. The new system achieved performance scores that were comparable to, and in key cases, superior to a standard Content-Based Filtering (CBF) model. Specifically, for test cases where the VGM atmosphere diverged from the game's core genre tags (e.g., the *Mario Kart 8* example), the audio-feature method achieved a $P@K$ of 0.88 and high subjective agreement, successfully recommending games based on shared audio characteristics across different genres. This validates that analysis into VGM provides a valuable and independent dimension for nuanced video game discovery.

The primary limitations encountered in this project involved data scarcity, as the dataset was restricted to VGM officially released on Spotify, and the inherent difficulty for a machine to objectively determine the context of when a soundtrack plays within a game.

Several ideas can be built upon this foundation for future work. Firstly, integrating Deep Learning models (such as Convolutional Neural Networks or Siamese networks) trained on spectrogram representations could potentially capture more complex, high-level musical features than RMS and MFCCs alone. Secondly, the scope could be expanded to include other audio characteristics, such as Chroma features and tempo, or by implementing a larger, multi-source dataset to mitigate the Spotify API limitation. Finally, a hybrid model that fuses the audio features with traditional metadata and collaborative filtering data is the most promising path toward a robust, production-ready recommender system for digital distribution services.

REFERENCES

[1] Cheuque, Germán, José Guzmán, and Denis Parra. "Recommender systems for online video game platforms: The case of steam." *Companion Proceedings of The 2019 World Wide Web Conference*. 2019.

[2] Lipscomb, Scott D., and Sean M. Zehnder. "Immersion in the virtual environment: The effect of a musical score on the video gaming experience." *Journal of Physiological Anthropology and Applied Human Science* 23.6 (2004): 337-343.

[3] Sun, Shiliang, and Rongqing Huang. "An adaptive k-nearest neighbor algorithm." *2010 seventh international conference on fuzzy systems and knowledge discovery.* Vol. 1. IEEE, 2010.

[4] McFee, Brian, et al. "librosa: Audio and music signal analysis in python." *SciPy* 2015 (2015): 18-24.

[5] Muller, Meinard, et al. "Signal processing for music analysis." *IEEE Journal of selected topics in signal processing* 5.6 (2011): 1088-1110.

[6] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." *Recommender systems handbook.* Boston, MA: springer US, 2010. 1-35.

[7] Panagiotakis, Costas, and Georgios Tziritas. "A speech/music discriminator based on RMS and zero-crossings." *IEEE Transactions on multimedia* 7.1 (2005): 155-166.

[8] Nagawade, Monica S., and Varsha R. Ratnaparkhe. "Musical instrument identification using MFCC." *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).* IEEE, 2017.

[9] Li, Qing, Sung Hyon Myaeng, and Byeong Man Kim. "A probabilistic music recommender considering user opinions and audio features." *Information processing & management* 43.2 (2007): 473-487.

[10] Mukhopadhyay, Sayak, et al. "Enhanced music recommendation systems: A comparative study of content-based filtering and k-means clustering approaches." *Revue d'Intelligence Artificielle* 38.1 (2024): 365.

[11] Silveira, Thiago, et al. "How good your recommender system is? A survey on evaluations in recommendation." *International Journal of Machine Learning and Cybernetics* 10.5 (2019): 813-831.

[12] Guo, Gongde, et al. "KNN model-based approach in classification." *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[13] Liu, Li-Ping, et al. "Transductive optimization of top k precision." *arXiv preprint arXiv:1510.05976* (2015).

[14] BharathiPriya, C., Akash Sreenivasu, and Sampath Kumar. "Online video game recommendation system using content and collaborative filtering techniques." *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA).* IEEE, 2021.