Feature Selection based Music Selection using Artificial Intelligence

Apoorva Bordoloi[1], Murari Prasad[1], Hem Thumar[1], Manas Saloi[1], Deepanshu Joshi[1], Shubham Mahajan[1,2], Laith Abualigah[3,4,5*]

[1]School of Engineering, Ajeenkya D Y Patil University, (iNurture Education Solutions Pvt. Ltd., Bangalore), Pune, India
[2]University Center for Research & Development (UCRD), Chandigarh University, Mohali, India
[3]Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al al-Bayt University, Mafraq 25113, Jordan.
[4]MEU Research Unit, Middle East University, Amman, Jordan
[5]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan.

Corresponding author: Aligah.2020@gmail.com

Abstract: Music recommendation systems have become increasingly popular in recent years thanks to artificial intelligence (AI). Streaming services have become increasingly popular in recent years thanks to affordable internet and media streaming services. Because the user base of music streaming services is ever-growing and the market in streaming services is also competitive, it is critical to provide listeners with recommendations to increase user base retention. Music recommendation systems have come a long way in the last decade, but there are still several issues to be addressed. Pure sound-based recommendations may have been a good option to examine. It would have been advantageous to explore pure sound-based recommendations since they would have been more accurate. The current systems available to consumers also have significant advantages, such as collaborative filtering of listeners, based on location, artists' preferred genres, etc., and recommending songs based on the results. Collaborative filtering does produce great results. This sound-based approach might achieve greater results when combined with these current methods.

Keywords: artificial intelligence, music recommendation, music streaming, sound-based recommendations, collaborative filtering

I. INTRODUCTION

Music recommender systems (MRSs) have become more and more popular as research in the field has expanded. Music streaming services like Spotify, Pandora, and Apple Music offer hundreds of millions of songs to music lovers. While MRSs can provide music that meets the patrons' requirements, they are still far from perfect and frequently generate unsatisfactory recommendations. This is because current MRS methods, which are usually focused on user-item interactions or content-based item descriptors, do not account for a user's musical preferences. We claim that to satisfy the customers' musical entertainment needs researchers and designers must consider three aspects of the listeners: intrinsic, extrinsic, and contextual. For example, psychologists and counsellors have demonstrated that people's musical interests and needs are influenced by factors such as age, gender, and sociocultural background. Because MRSs are almost always focused on user-item interactions, they do not sufficiently account for a user's musical preferences.

Within the music recommendation domain, we investigate three main issues namely, cold start, automatic playlist continuation, and evaluation of MRS, which are all to some extent prevalent in other recommendation systems. We also observe these issues to some extent in other recommendation systems, but certain characteristics of music make these domains particularly difficult. For example, music has a brief lifespan and a strong emotional connection, which makes it more challenging to recommend duplicates. In the second part of our discussion, we discuss future research initiatives. We investigate psychologically inspired music recommendations (in consideration of human personality and emotion) as well as situation- and culture-aware music recommendations.

A recommendation algorithm and software tool that connects to numerous real-world applications to give users the best choices for items they are most interested in. Recommendations for music selections, books to read, and music to listen to are connected to numerous real-world applications, such as what to buy, what music to listen to, and what current news to read. The music industry's transformation from commodity sales to subscriptions and streaming because of Apple's acquisition of Beats Music in 2014 resulted in a valuable digital music resource. Music providers must sell more diverse music as well as increase customer satisfaction by offering song selections that are most appropriate to users.

We can use lifestyle, age, gender, and other factors to identify a user's music preferences. To suggest music to users, we may use user profiling to determine their preferences. Audio metadata can include editorial, cultural, and acoustic metadata. We can model users with user modelling, which estimates the percentage difference in user profiles to determine their music preferences. For instance, Bogdanov et al used genre metadata to increase listener satisfaction. Collaborative filtering and content-based filtering are two methods for matching algorithm matching. Collaborative filtering relies on the assumption that users rate music items similarly or similarly in terms of usage, for them to rate other items similarly.

Because the assessments are harder to determine, collaborative filtering methods are more difficult to use because most users perceive only a small portion of all library material, making most assessments less prominent. On the other hand, content-based strategies determine music preferences using features of music items. "Content" in this context refers to the data in the files. Content-based approaches typically employ a two-step procedure, first extracting audio content features from audio material and then predicting user preferences by using them as a test. The extraction and comparison of audio characteristics such as timbre and rhythm have been the focus of a lot of research.

II. RELATED WORK

A good music recommender aims to help users filter and discover music based on their tastes, Y Song states in his paper published in 2012***[1].*** A good music recommender system should be able to automatically detect preferences and generate playlists accordingly. Furthermore, the development of recommender systems provides a great opportunity for the music industry to aggregate the users who are interested in music. We have to better understand and model people's preferences in music to develop recommender systems that are better able to satisfy them. In concert with the use of content-based modelling, the user can obtain lists of similar music based on acoustic features such as rhythm, pitch, or other fundamental features. A music recommender system is made up of three key components - users, items, and user-item matching algorithms. User profiling is used to distinguish users' tastes from each other. This step distinguishes users' tastes by using basic information. Item profiling, on the other hand, describes three different types of metadata - editorial, cultural, and acoustic, which are used in different recommendation strategies.

Xinxi Wang in his paper mentions that to achieve good content features for content-based music recommendation, a set of practical content guidelines must be assembled manually crafting such features is difficult, tiring, and time-consuming ***[2].*** A better approach is to combine the existing two-stage procedure with an automated process to learn features automatically, resulting in the creation of a unified and automated procedure: features are learned automatically and directly from the audio content. People have already begun employing deep learning to learn features for other music tasks such as music genre classification and music emotion detection with promising results. In existing content-based methods and hybrid methods, traditional features still play a significant role.

The authors in the paper *Deep content-based music recommendation* state that the characteristics of songs that influence user preference are difficult to ascertain from audio signals because much usage data is unavailable ***[3].*** To address this issue, a latent factor vector can be constructed that describes the differences in users' tastes and the corresponding characteristics of the items. It is often impossible to estimate these vectors, so they can be predicted from music audio content. The similarities between the characteristics of a song that affects user preference and the corresponding audio signals are great. A powerful model that captures the hierarchical, complex structure of music is required to extract high-level properties such as genre, mood, instrumentation, and literary themes from audio signals. Furthermore, the popularity of the artist, their reputation, and their location cannot be inferred from audio signals by themselves. The paper by Markus Schedl, talks about some grand challenges to be tackled in the development and evaluation of music recommendation systems ***[4].*** Challenges such as the duration of music have a high fluctuation from genre to genre and artist to artist. The volume or amount of music is also exponentially large. People usually hear music in a sequential order, where the order might also be of influence to some. Avoiding older recommendations to the same listener. Consumption habits

and listening intent such as duration, frequency, the purpose of listening, etc. Emotions influence the listener to play a major role in the music he/she likes. These challenges are the major limitations of recommendation systems. Adiyansjah, mention about the availability of features on music streaming applications like Spotify and Pandora to recommend music to users ***[5].*** These features can help users find music that fits their tastes by recommending music that has been previously enjoyed. This ensures that streaming music remains popular by keeping track of music that has been previously listened to. The music recommender system must be able to find appealing new music for all audiences, and it must do this in a way that matches the users' musical preferences. This is more complex than a general recommender system because the music-personalized recommender system must take into account the individual preferences of the user. As deep learning has developed, the results of deep neural networks have been promising in different fields, including music recognition. Mohamadreza state in their paper written in 2021 ***[6].*** They produced spectrograms from music pieces and scaled them on Mel-scale and created patches for all music pieces. The patches were fed to a convolutional neural network. They achieved better accuracy by combining acoustic and visual features. This paper also proposes an architecture for collecting required features (feature vectors) from intermediate layers of the CNN and then using Cosine similarity and Euclidean distance to classify music types. To have a feature vector with good quality, both max and average pooling are used. No filters have been used, such as collaborative filtering and user filtering. All of the decisions for recommending similar music are left to the system itself. Athulya K M talk about extracting meaningful spectrograms such as Spectral centroid, Spectral roll-off, Zero-crossing rate, Spectral bandwidth, and Chromo frequencies from slices of spectrograms such as Mel-spectrograms ***[7].*** Spectral centroid, Spectral roll-off, Zero-crossing rate, Spectral bandwidth, and Chromo frequencies are among the many rich features that spectrograms offer. It is a valuable activity for studying and understanding the connections between songs and genres. Using librosa Python library, we can extract the feature values from spectrograms.

III. METHODOLOGY

A. Autoencoders

An autoencoder is a neural network that learns to represent data by constraining the network's output. To do this, we will design a neural network architecture that restricts the flow of information through the network. Because each input feature is independent of all the others, compressing and re-creating the input data would be a difficult task if it were not for the structure that may be discovered in the data (correlations between input features). This structure can then be used to force the input through the bottleneck of the network.

B. Convolutional Networks

A Convolutional Neural Network can identify features in an image and adjust the weights and biases to learn to recognize them. In comparison to other classifiers, the processing required by a convolution network is significantly reduced. Because filters are hand-crafted in primitive methods rather than being constructed through training, they can recognize these filters/characteristics through training. The structure of a convolutional network is like that of the visual cortex of a human brain and is inspired by the organization of the visual cortex. A receptive field is in a small region of the visual field and responds to stimuli only. A collection of such receptive fields, which overlap the entire visual field, is known as the Visual Field. A Basic CNN consists of a Feature Extraction block of Convolution layers and Pooling layers, along with a Classification block of Fully Connected layers. Compared to other neural networks, the performance of convolutional neural networks is particularly impressive with image data, thus should perform well in our case of Mel-Spectrogram data.

C. Mel-Spectrogram

The mel-spectrogram is a condensed version of the spectrogram that is constructed by applying the Fourier transformation to overlapping time intervals in an audio snippet. The mel-spectrogram illustrates a compressed version of the spectrogram based on human perception. The mel-scale is developed through the observation that lower frequencies reveal more detail in human hearing ***[8-10].*** A compressed form of the mel-spectrogram has shown to be beneficial for deep-learning applications because it allows for a visual representation of audible features and frequencies in audio. Thus, allowing neural networks to learn these features, such as with the likes of a deep CNN.

IV. EXPERIMENT

A. Dataset

The GTZAN dataset was used ***[11].*** This dataset consists of 1000 music samples of 30 seconds each, with a total of 10 genres. We are aware of the problems and nuances of the GTZAN dataset, but firmly believe that it is still good enough to be used for working on our concept ***[12-14]***.

The audio files were in .wav format ***[15]***. We generated Mel-Spectrograms of all our 1000 music samples using the *librosa* package available in Python. The Mel-Spectrogram was generated with *n_mels=128* and *f_max=sr/2* (where sr is sampling rate of the audio clip). These Mel-Spectrograms were then saved with a resolution of 224 x 244 pixels which includes a white padding of 0.1 x screen dpi (96 in our case, thus 9.6 ≈ 10 pixels). The colour map used was *magma.* This process was done using *pyplot* library in matplotlib package for Python. The files were stored in .png format.

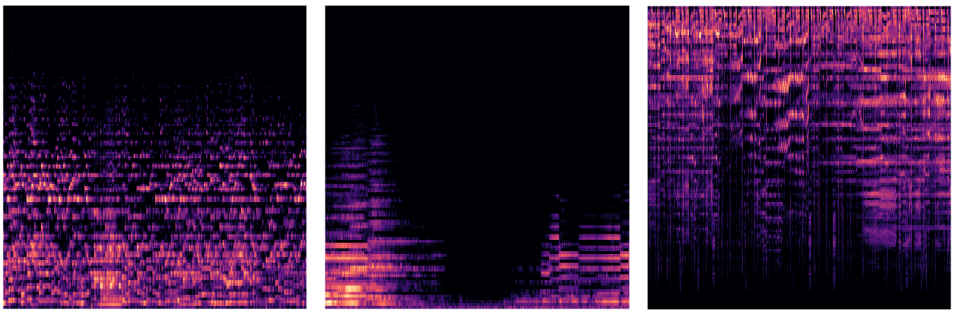

Figure 1: Few Examples of the Mel-Spectrograms

B. Model

We are using the pre-trained VGG-19 network available in Tensorflow for the encoder ***[16-19]***. The VGG-19 network consists of Convolutional layers followed by Fully Connected Layers. We removed the Fully Connected Layers which were used for class prediction in the original approach. Then the remaining network was partially training-locked by setting the layers.trainable parameter set to False, with only the last 5 layers set to True. By using transfer learning on VGG-19 we are saving training time, which results in better performance in most cases and also reduces the need of having a huge dataset. The decoder part was trained from scratch. The original VGG-19 has total 143,667,240 trainable parameters. After removing the fully connected layers in our encoder we are left with 20,024,384 total parameters. For our decoder we have 1,572,259 total parameters. The original VGG-19 is trained on ImageNet database that contains a million image of 1000 categories.

VGG-19 has 16 convolutional layers which are used for feature extraction and 3 fully-connected layers which are used for classification. The convolutional layers produces a tensor of size 7x7x512 for each image.We remove the last 3 classification layers. The layers used for feature extraction are segregated into 5 groups (or blocks). We lock these blocks and train only the last block of convolutional layers.

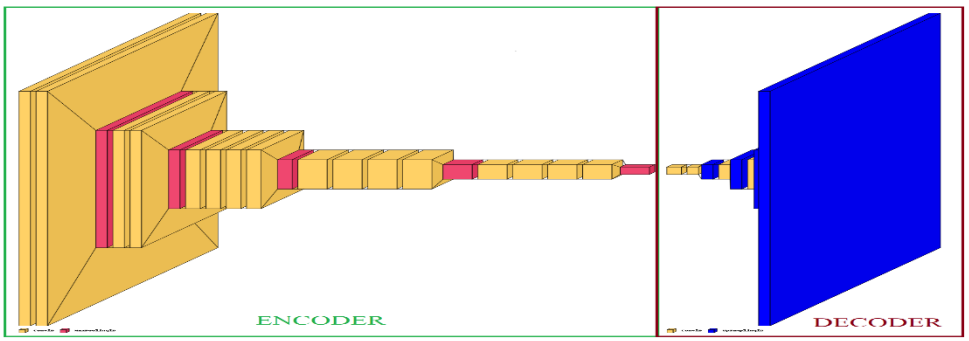

Figure 2: The autoencoder architecture representation. The Conv2D layers are depicted by yellow, MaxPooling2D by red and UpSampling2D by blue.

We combine both the encoders and decoders into an autoencoder. Autoencoder are commonly used for finding similar images in an unlabelled image dataset. Autoencoders compress the input data, and reconstructs it as an output. This makes autoencoders excellent for dimensionality reduction and helps us to focus on areas of

important features. Since we are working with spectrogram data, autoencoders help us in picking up the most pronounced parts of a spectrogram which can be used as a feature set.

C. Training

The data was split into training and testing set in the ratio of 90%:10% respectively. The autoencoder was then trained for 500 epochs using the Adam optimiser, with *lr=0.001* and *use_ema=True*. Mean Squared Error (MSE) was used as the loss metric for training along with Mean Absolute Error (MAE) as an additional metric just to monitor. The training loss, MSE was 0.0147, while we also achieved MAE of 0.0640. The testing set achieved MSE=0.0216 and MAE=0.0812.

Epochs	MSE	MAE
100	0.0179	0.0742
200	0.0165	0.0697
300	0.0155	0.0668
400	0.0148	0.0645
500	0.0147	0.0640

Table 1: Training MSE & MAE

V. RESULTS AND DISCUSSION

The reconstructed images showed that the Autoencoder manages to reconstruct our spectrograms highlighting the feature regions in the original images. The reconstruction has an overall blurriness to it due to the loss of low-level features during the encoding process. This was partially intentional as our motivation was to use this architecture to obtain encodings of high-level features of our spectrograms, which will later be used for our similarity checks. Now to select similar songs, first encodings for our dataset was obtained using the trained encoder. The encoder outputs a 7 x 7 x 512 matrix. This matrix is flattened for similarity measurement. Then songs were randomly selected with the motive being to find a song similar to every randomly selected song. Cosine Similarity & TS-SS Similarity, which has shown great performance, was used for measurement ***[20]***.
On measuring, the encodings managed to find a couple of similar songs with a Cosine Similarity of 0.6 - 0.7. TS-SS Similarity ranges from 0 to ∞, thus the encoding which managed the smallest TS-SS relative to the others, was selected as a similar song.

VI. CONCLUSION

From this experiment, we see it is possible to use a neural network such as an Autoencoder, which self-learns features, which can later be used as a representation of individual audio clips and compared with.
On further speculation of our experiment, we noticed that the songs which came up in our selection, often managed to overlap in genre. On listening to the audio, we usually found our songs to have an overlapping instrument or sounds, for example a fast drum region, or a loud bass line, etc. It seems as if the encodings did manage to capture the nature of the song and our similarity measurement is find of other encodings with similar nature. This is to be tested further in the future with more computational power and a larger dataset.
Furthermore, statistically testing the potency of our approach is also very difficult. Finding if two musical audio clips are similar or not is somewhat of a personal choice and changes from person to person. We can compare certain features which match and consider them to be similar, but still finding true similarity between music will require a more deep and extensive approach.

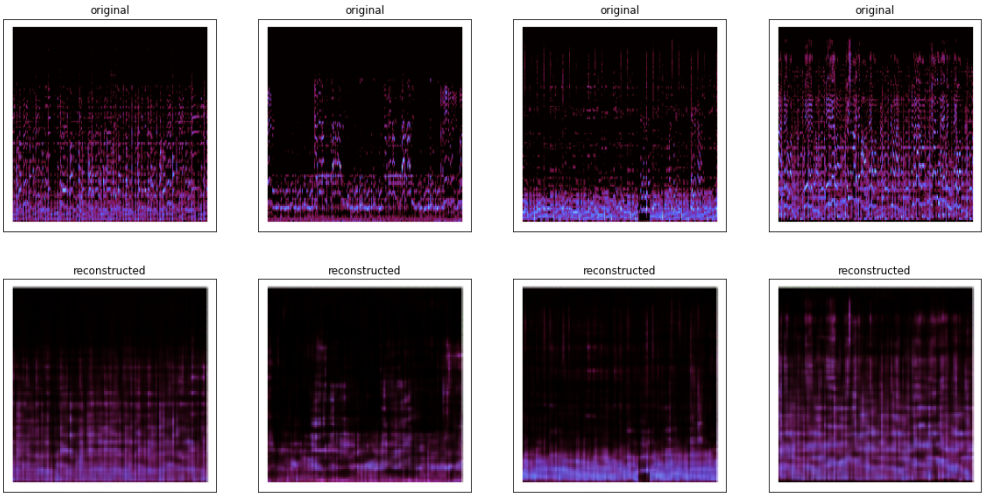

Figure 3: Original & Reconstructed Spectrograms

VII. FUTURE SCOPE

Unlike using predefined characteristics of a sound wave like pitch, frequency and wavelength, our approach made use of the patterns made by the waves themselves, and features were learnt autonomously. It would be interesting to see how this performs on a very large dataset which would represent a real-world scenario. Our motivation to use GTZAN over the other, more recent and larger datasets, was the fact that GTZAN had a representation of how real-world music and playlists are. From our research, the newer datasets we were able to find were more audio oriented and less music oriented.

It would also be exciting to test this with existing music recommendation solutions. Not in competition, but rather as an addition for improved recommendations. Us, the authors, have personally felt how lacklustre recommendations get. The current system just cannot do justice to the human nature of having a particular 'taste' in music.

REFERENCES

[1] Song, Yading, Simon Dixon, and Marcus Pearce. "A survey of music recommendation systems and future perspectives." *9th international symposium on computer music modeling and retrieval*. Vol. 4. 2012.

[2] Wang, Xinxi, and Ye Wang. "Improving content-based and hybrid music recommendation using deep learning." *Proceedings of the 22nd ACM international conference on Multimedia*. 2014.

[3] Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen. "Deep content-based music recommendation." *Advances in neural information processing systems* 26 (2013).

[4] Schedl, Markus, et al. "Current challenges and visions in music recommender systems research." *International Journal of Multimedia Information Retrieval* 7 (2018): 95-116.

[5] Gunawan, Alexander AS, and Derwin Suhartono. "Music recommender system based on genre using convolutional recurrent neural networks." *Procedia Computer Science* 157 (2019): 99-109.

[6] Sheikh Fathollahi, Mohamadreza, and Farbod Razzazi. "Music similarity measurement and recommendation system using convolutional neural networks." *International Journal of Multimedia Information Retrieval* 10 (2021): 43-53.

[7] KM, Athulya. "Deep Learning Based Music Genre Classification Using Spectrogram." *Proceedings of the International Conference on IoT Based Control Networks & Intelligent Systems-ICICNIS*. 2021.

[8] Pedersen, Paul. "The mel scale." *Journal of Music Theory* 9.2 (1965): 295-308.

[9] Singh, Ram, et al. "Impact of quarantine on fractional order dynamical model of Covid-19." *Computers in Biology and Medicine* 151 (2022): 106266.

[10] Ong, Song-Quan, et al. "Comparison of pre-trained and convolutional neural networks for classification of jackfruit artocarpus integer and artocarpus heterophyllus." *Classification Applications with Deep Learning and Machine Learning Technologies*. Cham: Springer International Publishing, 2022. 129-141.

[11] Tzanetakis, George, and Perry Cook. "Musical genre classification of audio signals." *IEEE Transactions on speech and audio processing* 10.5 (2002): 293-302.
[12] Sturm, Bob L. "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use." *arXiv preprint arXiv:1306.1461* (2013).
[13] Sharma, Shivalika, et al. "Image-based automatic segmentation of leaf using clustering algorithm." *International Journal of Nanotechnology* 19.6-11 (2022): 539-553.
[14] Mahajan, Shubham, et al. "Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks." *Soft Computing* 26.10 (2022): 4863-4881.
[15] https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification
[16] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
[17] Mahajan, Shubham, et al. "Fusion of modern meta-heuristic optimization methods using arithmetic optimization algorithm for global optimization tasks." *Soft Computing* 26.14 (2022): 6749-6763.
[18] Lakshmi, Yedida Venkata, et al. "Improved Chan algorithm based optimum UWB sensor node localization using hybrid particle swarm optimization." *IEEE Access* 10 (2022): 32546-32565.
[19] Singh, Harbinder, et al. "Performance evaluation of Non-Uniform circular antenna array using integrated harmony search with Differential Evolution based Naked Mole Rat algorithm." *Expert Systems with Applications* 189 (2022): 116146.
[20] Heidarian, Arash, and Michael J. Dinneen. "A hybrid geometric approach for measuring similarity level among documents and document clustering." *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2016.