FINAL REPORT

You Li
Student# 1004891511
youforest.li@mail.utoronto.ca

Luohong Wang
Student# 1005073313
luohong.wang@mail.utoronto.ca

Devansh Khare
Student# 1004980292
devansh.khare@mail.utoronto.ca

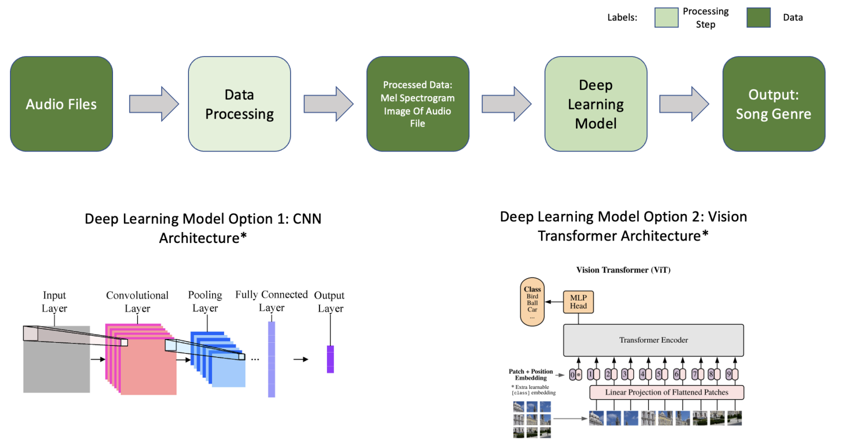
Rose Hyunjie Kim
Student# 1004341134
hjrose.kim@mail.utoronto.ca

1 INTRODUCTION

The purpose of this project is to create a tool that will be able to recognize the genre of a given piece of music based exclusively on the audio content. The motivation behind the project comes from the need for music recommendations. With the rising number of music streaming services, a personalized recommendation system attracts more users and boosts engagement (Dee, 2022). The platforms are then able to use the data generated by its additional users to further improve their model. This project will attempt to recreate genre classification which is one of the essential parts of music recommendation.

Machine learning is the ideal approach to this problem as there is a very large amount of unprocessed data that comes from various music sources with no specific ruleset for recommendation. A deep learning model can learn from these data points and make a prediction based on its training.

2 ILLUSTRATION/FIGURE


Labels:
Processing Step
Data
Audio Files
Data Processing
Processed Data: Mel Spectrogram Image Of Audio File
Deep Learning Model
Output: Song Genre
Deep Learning Model Option 1: CNN Architecture*
Input Layer
Convolutional Layer
Pooling Layer
Fully Connected Layer
Output Layer
Deep Learning Model Option 2: Vision Transformer Architecture*
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

*Input images will be mel spectrogram representations. This illustration only shows our inspirations for model architecture. Actual model architecture is shown in the 'Architecture' section of the report.

Figure 1: Overall Structure of the Project (Bisharad & Laskar, 2019) (Dosovitskiy et al., 2020)

Figure 1 describes the overall flow of data inputs and outputs that the neural network will be expecting for this project. It also graphically depicts how the creation of the depp learning model can be approached, through either CNN (convolutional neural network) or a Vision Transformer.

3 BACKGROUND & RELATED WORK

Popular song streaming applications use music sorting algorithms, such as Spotify. Spotify takes the user's song selection when they use the app as the live data input. NLP (Natural Language Processing) is used to categorize songs based on the emotion that it exhibits by searching keywords on the web that are relevant to the song (Wiggers, 2019). After the songs are categorized, Collaborative Filtering model recommends songs to individual users based on the music selection data collected. Particularly, Spotify's algorithm looks for subscribers who have similar personal playlists and listening history.

There have also been attempts to categorize songs by extracting exact song features using spectrogram image representation of the audio files. Researchers from KAIST categorized music by using the spectrogram of various songs as data inputs (Wiggers, 2019). The images were passed as the input through a Zero-Shot Learning model which is used in parallel with semantic search of various keywords that describe the music's genre and emotion. The keywords include a range of labels such as list of instruments in a song and types of genres. By doing so, without having the data set labeled, the model could identify the song genres and categorize it.

Methods used in music genre classification can also be applied to similar yet ever so different tasks, such as musical instrument classification. In a research paper that proposes a method to classify traditional Chinese instruments, there is a high emphasis on improving feature extraction techniques (He, 2022). Using MIDI files as data input, piano curtain matrix and Euclidean distance are used to generate Gaussian convolutional kernel. Novelty curve generated are used in parallel with BI-GRU to describe the sequential characteristics of the music. The segmented music is then passed through a backwards passing neural network to be classified.

Music genre classification has been tackled with many different types of neural network. An article suggests that there have been attempts in both academia and by individuals to classify music genre using CNN architecture, RNN, LSTM, and transformers (Vishnupriya & Meenakshi, 2018). In general, using purely a CNN architecture can output varying accuracy ranges from 40 to 70 percent. In another article, using the transformer model compared to the CNN model on the same task can be more effective at times; however, the accuracy level achieved is still variable and not as much of a significant improvement from CNN. (Khasgiwala & Tailor, 2021).

4 DATA PROCESSING

There were two datasets selected for training. The first one is selected from Kaggle, an online public source for users to explore, analyze and share data (kag). The dataset itself is the GTZAN Dataset - Music Genre Classification. It is the most used public dataset for evaluation in machine listening research for music genre recognition and considered a benchmark in the field (Olteanu, 2020)(mac). The GTZAN includes 10 genres of music, with 100 audios in each genre at 30 seconds each. The second dataset is selected to be a much larger dataset in order to see the effects on training. The dataset is named the fma_small. It is used by The International Society for Music Information Retrieval (ISMIR) and contains 8 genres with 1000 clips of 30 seconds each. This data set is taken from an open source GitHub repository (Defferrard et al., 2017). However, upon training of the larger dataset, the group realized that the quality and classification of audio within this dataset is significantly worse and thus yielded much worse results compared to the GTZAN. For the purpose of this report, only the GTZAN will be covered beyond Data Processing.

To process the datasets, the group had to first clean it in a way that can be used for the models. For GTZAN, this was relatively simple as the audio files are split into their categories. However, for the fma dataset, the files are presented in a large number of seemingly random folders. To identify their genres, a metadata CSV file was provided and had to be looked through in order to sort the data. After the cleaning, the group decided to further split the data into 3 second clips, increasing the data points by a factor of ten to compare training results. The data is then split into 85-15 train and validation, with testing being selected manually and discussed in later sections of the report.

Before the data can be fed to train a model, a way to represent the data through cleaning and formatting is required. There are several features that may be taken into consideration given a piece of audio file. An audio wave is made up of amplitude, frequency and time. The amplitude and time can be acquired through sampling of the audio data, afterwards the Fast Fourier Transformation (FTT) can then be used to extract individual frequency waves from the consolidated audio wave (Velayudham, 2020). However, since the human hearing range is only between 20Hz to 20kHz, waves that fall outside of the range should not be considered as heavily for the analysis. The Mel Frequency Cepstral Coefficients (MFCC) is a non-linear transformation scale where it adds special consideration to a human's hearing range value (Nair, 2018). Since the genre of a piece of music largely depends on how a human perceives the music, MFCC is the ideal operation to be conducted on the given dataset. The MFCC formula from source: $Mel(f) = 2595 \log(1 + \frac{f}{700})$ (Nair, 2018).

The MFCC graph would then be combined with tempo to produce images that will be processed by the model. Although the Kaggle data source provides the images of Mel Spectrogram representation of each of the audio files for GTZAN, the data collecting and processing would still be done on the team's side to ensure consistency within the project. An example of a Mel Spectrogram is shown below.

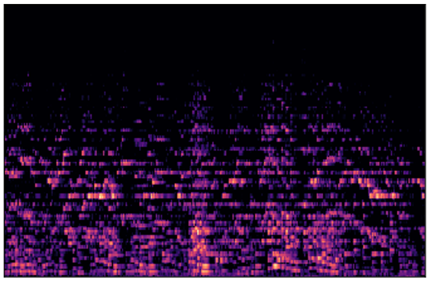


Figure 2: Mel-Spectrogram of a 30 second country music clip

Following the image generation, the team has also decided to extract certain features from each of the audio files. The features extracted include: Short-time Fourier transform (STFT) of chromagram, root-mean-square (RMS) value, spectral centroid, spectral bandwidth, roll-off frequency, zero-crossing rate, Mel-frequency cepstral coefficients across 20 channels, harmonic elements, percussive elements and tempo. The mean and variance of each of the features were selected in order to normalize the data.

5 Architecture

The first type of neural network that the group used is CNN, which is a common neural network used in the field of deep learning. The CNN first uses a convolution layer consisting of element-wise multiplication with a kernel layer. Following the convolution layer is a max pooling layer, which is used to reduce the dimension of the feature map. Since the kernel layer is shared across the layer, it can be useful for music genre prediction, as images such as Mel Spectrograms exhibit unique variations in its pattern across different music genres. The hyperparameters of the network include the number of hidden layers, kernel sizes, padding sizes, batch size, learning rate, activation function and loss function.

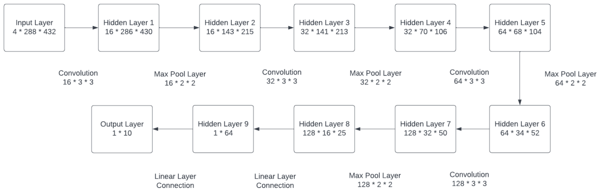
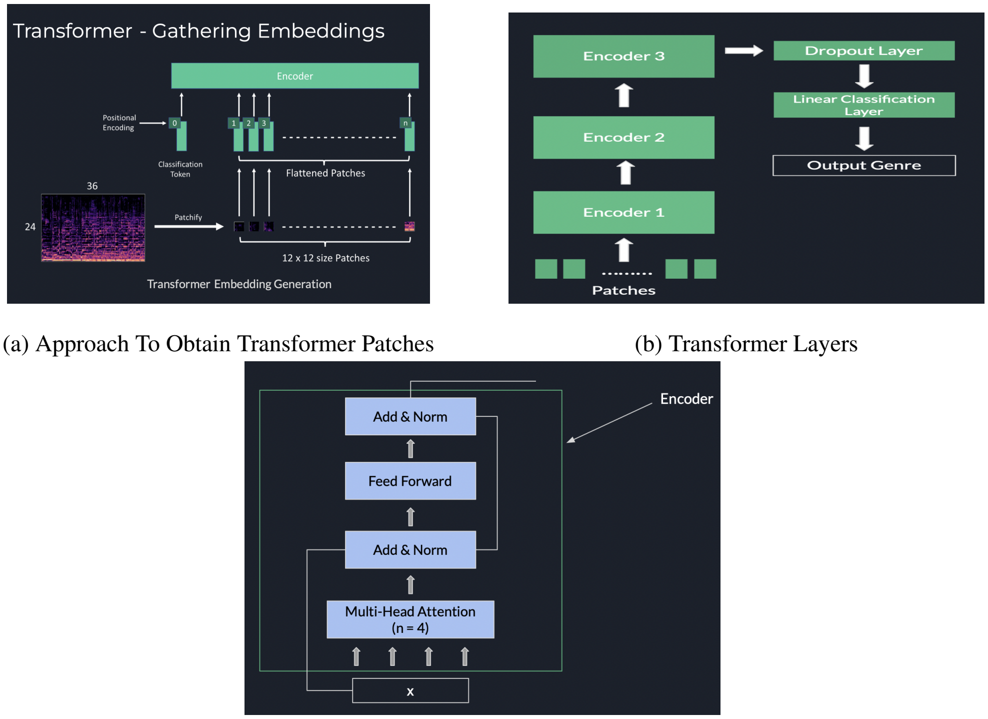


Figure 3: CNN model group used for training

Another deep learning model that the group explored in this project is a vision transformer. A transformer's advantage lies in being able to compute multiple patches in parallel, allowing theoretically faster computations than other sequential neural networks. Vision transformers in particular divide images into "patch tokens", compute the embeddings for each patch token, add positional embeddings to retain temporal relationships and then process the token embeddings through an encoder to classify the output. In the case of the mucic recognition model, the Mel Spectrogram images of an audio clip are used as an attempt to classify them to the right genre using the vision transformer. The transformer structure has been shown in the figures below.



(a) Approach To Obtain Transformer Patches

(b) Transformer Layers

(c) Encoder Design

Figure 4: Transformer Architecture

To minimize overfitting by the model, the model incorporated L2 Weight Decay, Dropout, Batch Normalization and Early Stopping With Patience during training. The following hyperparameters were used to get the highest validation accuracy using our vision transformer model:

Table 1: Selected Hyperparameters For Best Transformer Model

Hyperparameter	Selected Value
Learning Rate	0.001
Batch Size	32
Epochs	20
Optimizer	Adam
Momentum	0.5
Regularizer	0.001
Dropout	0.25

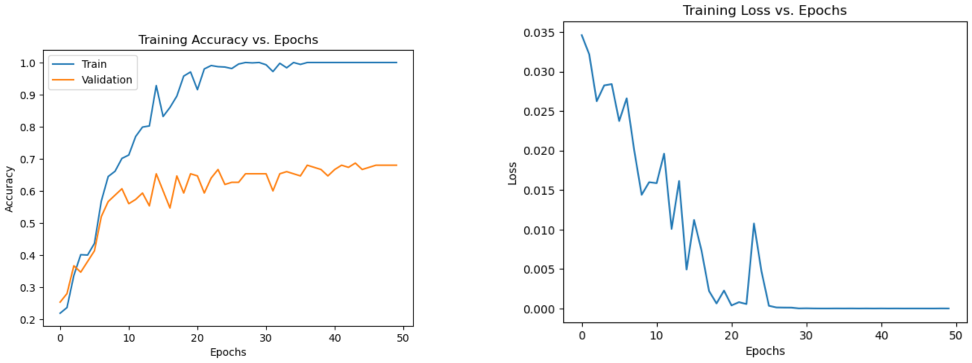
6 Baseline Model

The baseline model that was selected is the KNN model, where an object is classified through a vote of its K's nearest neighbours. KNN is an appropriate baseline model for Music Recognition since similar music features such as tempo, harmonics, percussive elements, etc, are good indications that those music are in the same genre. The dataset used for KNN model is the features CSV file extracted from data processing. KNN classification was performed on both the 30 second version and the 3 second version to understsnd how the data differs. For each data set, the models were ran on 10 neighbours, 20 neighbours and 100 neighbours. With the 3 second version, the validation accuracy was 86.8%, 81.8% and 69.2% respectively, while the 30 second version's accuracy was 62.7%, 57.3%, and 44.7% respectively. As expected, the KNN model done on the 3 second CSV file shows much better accuracy on the validation set compared to the 30 second model. The KNN model that yielded the best result in nearest neighbour value of 10, with validation accuracy of 0.868.

7 Quantitative Results

7.1 CNN Model

The model had a final training loss of 0.0080, final training accuracy of 100%, and final validation accuracy of 68%. The figures depicting the same have been attached for the model below:

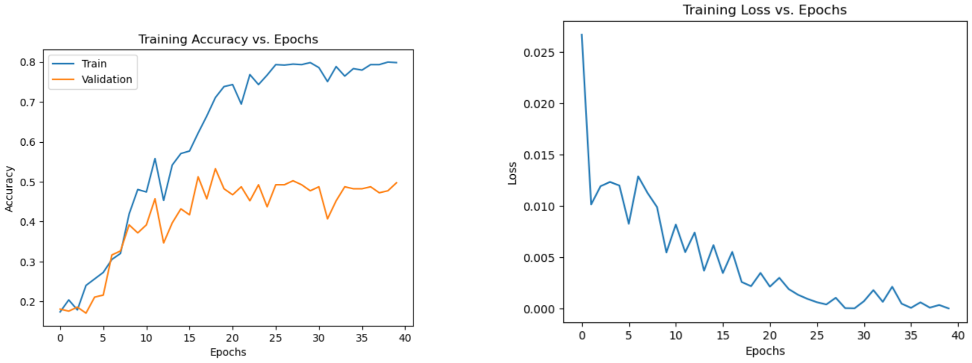


(a) 30 second GTZAN Data accuracy

(b) 30 second GTZAN Data loss

Figure 5: CNN Model Quantitative Results

Similar model was used for the GTZAN data set with splitting. With the larger data set, the model yielded final training loss of 0.00003268, final training accuracy of 79.8% and final validation accuracy of 49.7%. The figures depicting the same have been attached for the model below:



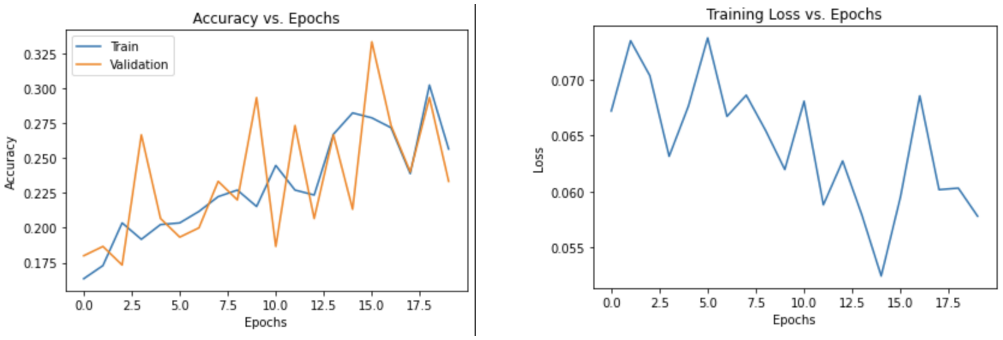
(a) 3 second GTZAN Data accuracy

(b) 3 second GTZAN Data loss

Figure 6: CNN Model Quantitative Results

7.2 Transformer

The transformer model had a final training loss of 0.058, final training accuracy of 25.64%, and final validation accuracy of 23.33%. The figures depicting the same have been attached for the model below:



(a) 30 second GTZAN Transformer Accuracy

(b) 30 second GTZAN Transformer Loss

Figure 7: Transformer Quantitative Results

These accuracies are significantly lower than the CNN model. Additionally, the transformer also does not yield significant advantage in running times over CNN despite its theoretical faster performance due to parallel processing.

Given the transformer's low performance, we can conclude that CNNs are the better approach for our application. Consequently, while the low validation accuracy for transformers will be explained in the Discussion section, the remaining sections of this report will focus on the CNN model's performance.

8 Qualitative Results

Since transformer model yielded low accuracy, qualitative results will be shown for just the CNN model. In the below table and graph, the accuracy per genre for the trained CNN model is listed.

Table 2: Validation Accuracy Per Genre

Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
40%	73.33%	53.33%	73.33%	13.33%	80%	60%	66.67%	66.67%	46.67%

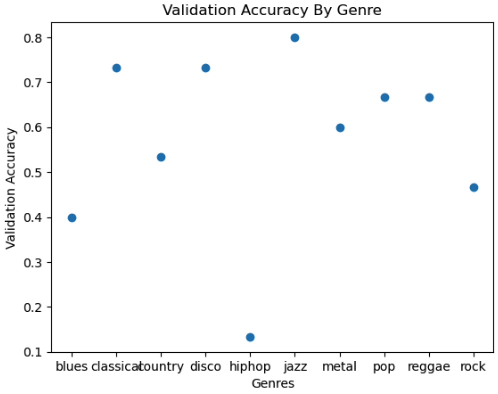


Figure 8: CNN model accuracy per genre

9 Evaluate Model on New Data

To ensure that the testing data has never been seen by any of the models, the team has decided to manually select testing music clips that will be used for testing. In order to achieve this, individual music clips per genre is downloaded from a free music archive, and each sample of 30 second clip is selected around the middle of each song (roy). To ensure that the music is not part of the GTZAN or fma_small, all the music selected for testing are created after the creation date of either of the datasets. The Mel Spectrograms of these clips were then generated for testing.

With new data provided, the CNN model yields a testing accuracy of 56.67%. This is relatively low compared to some of the better models currently done by the researches. Multiple factors may contribute towards the low accuracy. The CNN model may not be complex enough to learn all the features within the Mel Spectrograms given per genre. A four layer neural network may still be too simple for the CNN to fully learn all the features. Furthermore, given only 1000 data points, 10 genres meaning each genre would only have 100 images split into training and validating, such small training sample size could easily lead towards over-fitting, as seen in the Quantitative Results section.

10 Discussion

The project focused on evaluating which deep learning approach would lead to the most effective classification. It was observed through testing and tuning that despite their theoretical advantages, transformers are a poor fit for the data available, leaving CNNs as the most effective model for music genre classification.

10.1 CNN Model

From the quantitative results, the CNN model does not yield a very high accuracy in both the validation set and training set. There are a few causes that could have lead to this result. Since the training accuracy reached 1.0, it is safe to assume that the model has over-fitted the data set. Despite efforts to reduce over-fitting such as weight decay and dropout, the model seems to still over-fit within small training cycles with little gain in the validation accuracy peaking at around 65% to 70%.

The qualitative results show over-fitting on the Jazz genre with accuracy of 80%, as well as under-fitting in genres blues, country, hip hop and rock with accuracy 40%, 53.33% 13.33% and 46.67%. This could be due to the similarities between the Mel Spectrograms generated for musics within each genre. If two music from different genres share similar features such as tempo, loudness and frequency, then the CNN model could have treated these two genres as the same genre, leading towards over-fitting towards one genre.

Lastly, the GTZAN data set with splitting yielded lower accuracy compared to the GTZAN data set without splitting. The group believe that this happened because of two reasons. First is that the splitted images share high resemblances with each other, and if the genre's share similar features as mentioned above, having more data points would only make the CNN model worse at differentiating between the genres. Secondly, the larger Mel Spectrogram images have a resolution size of 480 * 640, which is larger than the average image resolution sizes used in lecture and labs with was 224 * 224. Having a larger resolution means it will be more difficult for the CNN model to learn all the different features among each genre.

10.2 Transformer

There are a few reasons that the transformer did not perform up to the group's standard. At first, the group had thought that the smaller GTZAN dataset causes the transformer to overfit quickly with an increasing number of parameters. This limits the ability to add more parameters, which is a shortcoming given that transformers inherently need more parameters for effective modelling. However, when the group attempted the transformer model with the larger fma_small dataset, it was evident that the fma_small has strong overlaps between the genres, which prevents the model from training effectively. Additionally, CNNs have been shown in research to be more parameter-efficient than CNNs when it comes to accuracy, which explains why for limited parameters, our CNN model outperforms our transformer.

11 Ethical Considerations

Ethical considerations can arise due to the input data set being music, which have copyrights. Aside from using existing audio dataset as mentioned above, GTZAN dataset, the selection of the test data set may require ethical considerations. The team is to agree that the neural networks developed for this project will not be used for commercial purposes and will be using publicly available audio files to test the data.

12 Project Difficulty / Quality

Overall, the group felt sufficiently challenged given the time and resources.

For data processing, the group experienced difficulties in dealing with the signal processing section, since none of the team members had prior knowledge and a large amount of research was needed. The group also had issues with the large quantity of data that needed to be cleaned and processed. This was especially prominent in the fma_small dataset. The lack of quality data also presented to be an issue. When the group attempted to use the larger fma_small dataset, it was evident that the worse quality of audio classification deterred the models from performing up to standard.

For the CNN and transformer models, it was challenging to tune the large amount of hyperparameters that the model possesses. Between number of convolution layers, convolution output layers, kernel sizes, stride sizes, padding sizes, dropout rate, weight decay, optimizer, loss function etc, finding the correct combination of hyperparameters makes the CNN tuning tasks very difficult and time consuming. Each tune will require a rerun of the entire model on the large amount of data, taking upwards of 20 hours per run and caused drastic differences. These factors led to limited options for the group within the allocated time to present a good model.

To conclude, the group has faced various difficulties with multiple elements of this project and a substantial amount of time and effort was spent trying to understand and solve these issues. The group has a unanimous feeling that the performance of the models satisfied the expectations, despite the challenging aspect of the project.

13 Link to Colab Notebook

https://colab.research.google.com/drive/1bEnPoQ-BnN6fZegwO2qujM_doygWvkZn?usp=sharing

REFERENCES

Datasets documentation. URL https://www.kaggle.com/docs/datasets.

Gtzan genre dataset. URL https://datasets.activeloop.ai/docs/ml/datasets/gtzan-genre-dataset/.

Fma - powered by tribe of noise. URL https://freemusicarchive.org/.

Dipjyoti Bisharad and Rabul Hussain Laskar. Music genre recognition using convolutional recurrent neural network architecture. *Expert Systems*, 36(4), 2019. doi: 10.1111/exsy.12429.

Catherine Dee. How do music recommendations work to boost user engagement?, Jun 2022. URL https://www.algolia.com/blog/ux/how-does-a-music-recommendations-algorithm-work-to-boost-user-engagement/.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. URL https://arxiv.org/abs/1612.01840.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Qi He. A music genre classification method based on deep learning. *Advanced Aspects of Computational Intelligence and Applications of Fuzzy Logic and Soft Computing*, 2022, Mar 2022.

Yash Khasgiwala and Jash Tailor. Vision transformer for music genre classification using mel-frequency cepstrum coefficient. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 2021. doi: 10.1109/gucon50781.2021.9573568.

Pratheeksha Nair. The dummy's guide to mfcc, Jul 2018. URL https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd.

Andrada Olteanu. Gtzan dataset - music genre classification, Mar 2020. URL https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.

Vasanthkumar Velayudham. Audio data processing- feature extraction-science amp; concepts behind them, May 2020. URL https://medium.com/analytics-vidhya/audio-data-processing-feature-extraction-science-concepts-behind-them-be97fbd587d8.

S Vishnupriya and K. Meenakshi. Automatic music genre classification using convolution neural network. *2018 International Conference on Computer Communication and Informatics (ICCCI)*, Jan 2018. doi: 10.1109/iccci.2018.8441340.

Kyle Wiggers. Ai classifies songs from genres it has never heard before, Jul 2019. URL https://venturebeat.com/ai/ai-classifies-songs-from-genres-it-has-never-heard-before/.