Project Proposal
[Music Genre Classification]

Group 96

So Hee Yoon	1004327604
Ho Seok (David) Lee	1004112177
Myeong Hun (David) Song	1004815961
Ting Wei (Sherman) Lin	1004835413

Word count: 1317

Penalty: 0%

Introduction

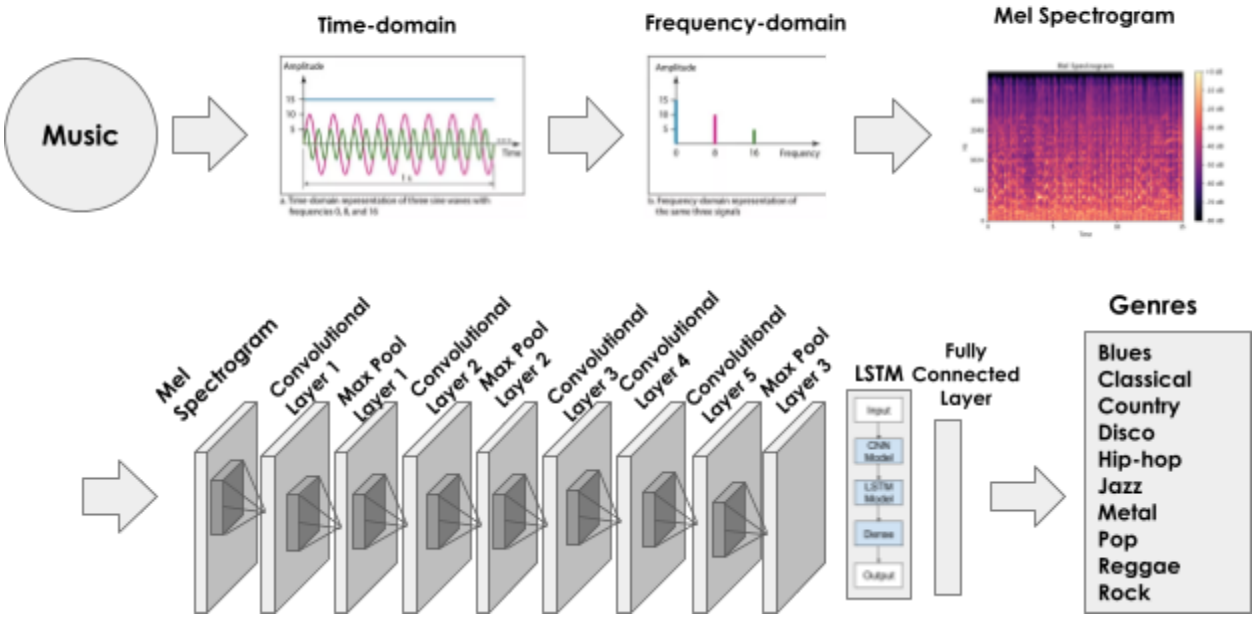
Music genre classification, unlike song recognition, is often thought of as a subjective decision. Although a human may be able to tell the difference between a blues song and a classical piece, writing an objective criteria on what defines a particular genre can be difficult. For example, metal and rock involve the same instruments and similar timbres, but a child would be able to tell that they are two distinct genres. Due to these characteristic subtleties, machine learning would be appropriate.

A solution to this problem can also be understood as the grouping of similar sounding songs. A “genre” is nothing more than a label for these groupings. Expanding upon this model, music streaming services such as Spotify and Apple Music would be able to apply this model to a much larger music dataset and improve their artist and song recommendations to users.

Illustration

[1] [2]

Figure 1. Basic Idea of our project



Background & Related work

Analyzing lyrics and chords to classify music genre by Timothy Greer and Shrikanth Narayanan [3]

This research uses natural language processing to classify music from the Billboard charts and to represent the relationship between chord progression and lyric sequence in predicting genre of the music.

Music genre classification by Derek A. Huang [4]

Convolutional Neural Network used to predict genres of the music, out of 10 distinct music genres. The model inputs two different data; raw amplitude data and their transformed mel-spectrograms to determine which input resulted in higher accuracy.

Data processing

Source of Data

Our dataset consists of 10 genres with 100 30-second music clips in each category, for a total of 1,000 songs. This dataset is also known as the GTZAN dataset [5].

Preprocessing

Step 1) Normalizing Data

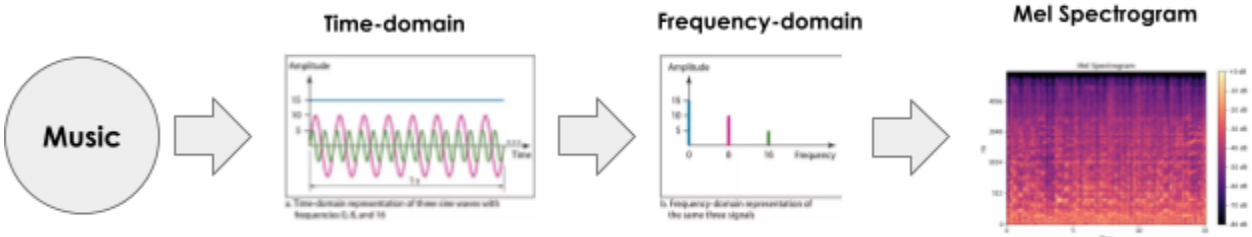
- We will resample the data to a sample rate, fixed for consistency, to reduce a continuous signal into many discrete values. The sampling rate decides the number of samples taken over a period [6]. We will use a sampling rate of 20050Hz, as it is popular for audio files [2].
- Using PyTorch library torchaudio:
    1) Open audio files (data sets)
    2) Use torchaudio.transforms.Resample() to resample data

Step 2) Create a Mono Audio Signal using torch.mean() [2].

Step 3) Transforming Audio Signal to Mel Spectrogram Image

- A spectrogram is a visual representation of the frequencies of a signal over time. A mel-spectrogram is a spectogram mapped to the mel scale, which converts the frequencies to a linear scale within the average human hearing range [2].
    1) Plot waveform
    2) Transform the waveform into mel spectrogram using torchaudio.transform.melSpectrogram()

Figure 2. Data processing procedure



Architecture

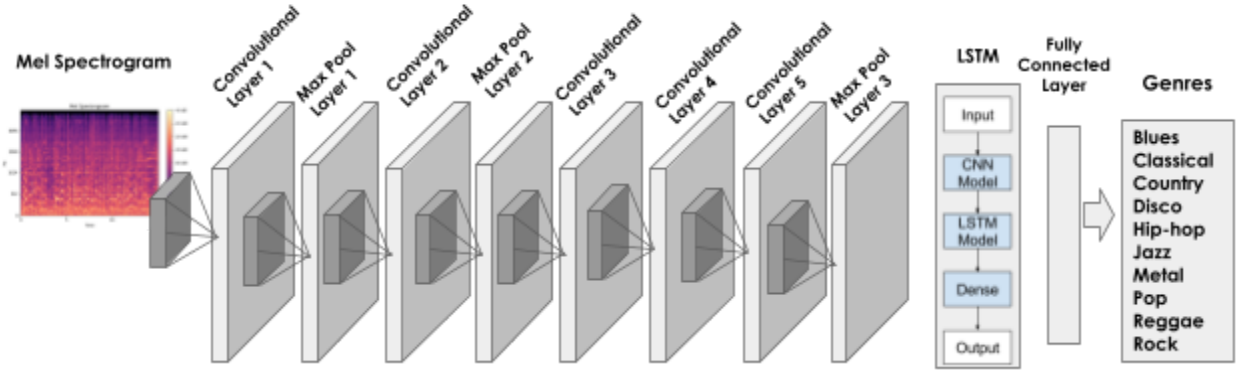
Since each audio input will be processed into a mel-spectrogram frequency graph [8], the required solution has now changed from an audio classification model to an image classification model. We will use a convolutional neural network (CNN).

Following the concept from the Visual Geometry Group (VGG) architecture, we will almost exclusively use 3×3 convolution kernels, but increase the depth of the network [7].

Although the input spectrogram is a single image, it still represents a sequence of sounds and frequencies over time. A long short-term memory (LSTM) module, a type of recurrent neural network (RNN), can help optimize sequential data, including audio and speech [8]. These LSTM modules adapt to the sequential nature of information in the audio, avoiding vanishing gradients. We will implement the LSTM module using Keras [9].

Our final convolutional neural network will consist of 5 convolutional layers and 3 max pooling layers, an LSTM module, and a final set of fully-connected layers for output classification. Similar architectures have been used for speech recognition and audio processing, and we believe its function can extend to genre classification.

Figure 3. CNN architecture



Baseline model

To compare the effectiveness of our model, we will compare the accuracy with a few rudimentary supervised baseline models. The two baseline models that we will be using are the k-nearest neighbours algorithm (k-NN) and a support-vector machine (SVM). Both of these models will be implemented in the scikit-learn Python library [10][11].

k-Nearest Neighbours Algorithm

The k-nearest neighbours algorithm is a classification algorithm that processes labeled data [12]. Given an unlabeled data point, the algorithm finds the closest labeled data point within the parameter vector space.

Each mel-spectrogram from the labeled training data will be converted into 20 mel-frequency cepstral coefficients (MFCC) to populate the vector space. For each unobserved datum, using the Minkowski distance

$$dist(x,\ z)\ = \left(\sum_{r=1}^{d} \left| x_r - z_r \right|^p \right)^{1/p},$$

the algorithm will find 5 nearest neighbours (i.e. closest sounding clips). The predicted label will be the most commonly occurring label from the 5 neighbours, weighted by their distance.

Support Vector Machine

A support vector machine attempts to maximize the separation between clusters of labeled vectors [13]. Since we do not know the general shape and behaviour of our data, we can blindly assume that the classes are not linearly separable. Changing the SVM kernel from a linear function to the Gaussian radial basis function (RBF)

$$rbf(x,\ z)\ = exp\left(-\ \gamma||x\ -\ z||^2\right),$$

allows the vector space to be mapped to an infinite dimensional feature space for the SVM to create non-linear decision boundaries. Once trained, the SVM can begin predicting the labels of unobserved data.

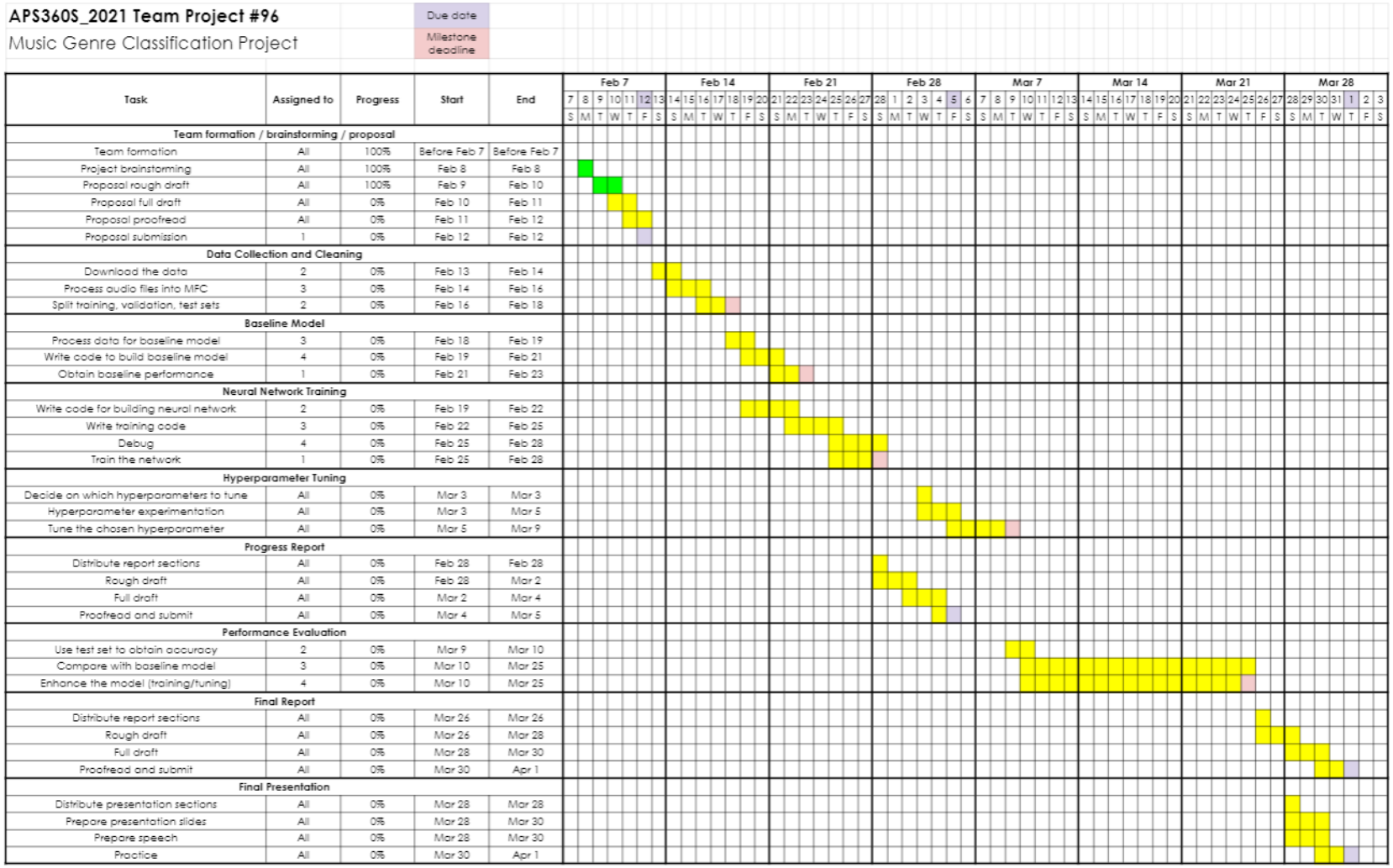
Ethical Considerations

Music is a form of artistic and creative expression. There isn't a definite description of how a certain genre of music has to sound like. A song can be very dynamic, composed of elements from many genres of music. By trying to give a definite answer and classifying a song to be only one genre of music, we are restricting its artistic expression. This puts a preconceived notion on a song that may go against the ideas of its creator. For example, our model may classify a song to be Rock music, but its author may claim otherwise.

Most music streaming platforms have playlists of different music genres. If a song is classified as only one genre and is only placed in one of the many genre-specific playlists, it will lose many potential listeners who don't listen to other playlists because of its labeled genre. As a result, an artist may lose profits on music royalties or the reach of a bigger audience. Furthermore, there is a potential for our project to replace jobs in the music industry that involves music classification, like playlist curators on music streaming platforms.

Project plan

Meeting Time	● There's no fixed time for meeting, due to different time zones we are at ● Team members select their available time When2Meet and meetings held when all members are available.  **Figure 4.** Example of when2meet usage for team meeting date and time selection
Team communication	● Facebook Messenger group chat ● Zoom for online meetings ● Discord for working sessions
Communication with TA	● Asking questions during lab session ● Email ● Private Piazza post
Document Writing	● Deliverables - Google Docs/Slides ● Storage - Shared Google Drive Folder
Programming	● Google Colab will be used: ○ can track the contribution of each team member ○ enables us to create multiple code sections ■ Code overwriting can be prevented
Timeline and Task Division	● Ultimate goal: finish work in week 10 (week before final presentation and report due) ● Project is divided into multiple milestones with internal deadlines ○ assigned with numbers representing team member ■ To be determined (to increase flexibility) ● Refer to the Gantt Chart for detailed outline



Task	Assigned to	Progress	Start	End
Team formation / brainstorming / proposal				
Team formation	All	100%	Before Feb 7	Before Feb 7
Project brainstorming	All	100%	Feb 8	Feb 8
Proposal rough draft	All	100%	Feb 9	Feb 10
Proposal full draft	All	0%	Feb 10	Feb 11
Proposal proofread	All	0%	Feb 11	Feb 12
Proposal submission	1	0%	Feb 12	Feb 12
Data Collection and Cleaning				
Download the data	2	0%	Feb 13	Feb 14
Process audio files into MFC	3	0%	Feb 14	Feb 16
Split training, validation, test sets	2	0%	Feb 16	Feb 18
Baseline Model				
Process data for baseline model	3	0%	Feb 18	Feb 19
Write code to build baseline model	4	0%	Feb 19	Feb 21
Obtain baseline performance	1	0%	Feb 21	Feb 23
Neural Network Training				
Write code for building neural network	2	0%	Feb 19	Feb 22
Write training code	3	0%	Feb 22	Feb 25
Debug	4	0%	Feb 25	Feb 28
Train the network	1	0%	Feb 25	Feb 28
Hyperparameter Tuning				
Decide on which hyperparameters to tune	All	0%	Mar 3	Mar 3
Hyperparameter experimentation	All	0%	Mar 3	Mar 5
Tune the chosen hyperparameter	All	0%	Mar 5	Mar 9
Progress Report				
Distribute report sections	All	0%	Feb 28	Feb 28
Rough draft	All	0%	Feb 28	Mar 2
Full draft	All	0%	Mar 2	Mar 4
Proofread and submit	All	0%	Mar 4	Mar 5
Performance Evaluation				
Use test set to obtain accuracy	2	0%	Mar 9	Mar 10
Compare with baseline model	3	0%	Mar 10	Mar 25
Enhance the model (training/tuning)	4	0%	Mar 10	Mar 25
Final Report				
Distribute report sections	All	0%	Mar 26	Mar 26
Rough draft	All	0%	Mar 26	Mar 28
Full draft	All	0%	Mar 28	Mar 30
Proofread and submit	All	0%	Mar 30	Apr 1
Final Presentation				
Distribute presentation sections	All	0%	Mar 28	Mar 28
Prepare presentation slides	All	0%	Mar 28	Mar 30
Prepare speech	All	0%	Mar 28	Mar 30
Practice	All	0%	Mar 30	Apr 1

Figure 5. Gantt chart, rotated

Risk Register

Risks	Solutions
Model takes too long to train	● Reduce time for the clip of each music ○ results in reduced input size ● Reduce number of genres ○ model will be trained to classify music into less labels ● Adjust hyperparameter ○ number of epoch ○ learning rate
Not enough data	● Data augmentation ○ stretch 30 seconds of audio clip into a minute than making into 2 separate data, without changing the pitch ● Split one 30 seconds audio clip into: ○ 6*5 seconds ○ 3*10 seconds
Music can be classified into more than one genre	● SoftMax activation function at the output layer of the model ○ normalize the outputs and give a genre as output
High complexity of the code	● Switch from CNN to ANN and extract MFCC feature from audio files, instead of MFC ○ the model will take numbers as input rather than the image

Link to the project

Google Colab:
https://colab.research.google.com/drive/1g3ETP0yloTeWH1OKEKVu5EZ1Yuh1abzU?usp=sharing

References

[1] Z. W. Engel, M. Kłaczyński, and W. Wszołek, “A Vibroacoustic Model of Selected Human Larynx Diseases,” International Journal of Occupational Safety and Ergonomics, vol. 13, no. 4, pp. 367–379, 2007.

[2] L. Roberts, “Understanding the Mel Spectrogram,” Medium, 14-Mar-2020. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53. [Accessed: 12-Feb-2021].

[3] T. Greer and S. Narayanan, “Using shared vector representations of words and chords in music for genre classification,” SMM19, Workshop on Speech, Music and Mind 2019, 2019.

[4] D. A. Huang, S. A. Arianna, and P. J. Eli, “Music Genre Classification,” 2018.

[5] Index of /sound/genres. [Online]. Available: http://opihi.cs.uvic.ca/sound/genres/. [Accessed: 12-Feb-2021].

[6] N. S. Chauhan, “Audio Data Analysis Using Deep Learning with Python (Part 1),” KDnuggets, Feb-2020. [Online].

[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Visual Recognition,” Visual Geometry Group - University of Oxford. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/research/very_deep/. [Accessed: 12-Feb-2021].

[8] B. N. Kaushik, “A Hybrid Technique using CNN+LSTM for Speech Emotion Recognition,” International Journal of Engineering and Advanced Technology, vol. 9, no. 5, pp. 1126–1130, 2020.

[9] J. Brownlee, “CNN Long Short-Term Memory Networks,” Machine Learning Mastery, 14-Aug-2019. [Online]. Available: https://machinelearningmastery.com/cnn-long-short-term-memory-networks/. [Accessed: 12-Feb-2021].

[10] “1.4. Support Vector Machines¶,” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed: 12-Feb-2021].

[11] “sklearn.neighbors.NearestNeighbors¶,” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html#sklearn.neighbors.NearestNeighbors. [Accessed: 12-Feb-2021].

[12] “Lecture 2: k-nearest neighbors / Curse of Dimensionality,” Cornell Computer Science - CS4780. [Online]. Available:

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html. [Accessed: 12-Feb-2021].

[13] R. Berwick, “An Idiot’s guide to Support vector machines (SVMs),” in 6.034 Artificial Intelligence - MIT.