Music Subgenre Classification with Deep Neural Networks

Cadence Kirby and Jaydev Bhateja
University of Washington, Seattle
3800 E Stevens Way NE, Seattle, WA 98195
ckirby03@cs.washington.edu, jbhateja@cs.washington.edu

Abstract

Broad music genre classification using neural networks has achieved high accuracy on standard datasets such as GTZAN. However, fine-grained classification between closely related subgenres—such as bluegrass, country, and folk in Western music and a variety of raags in South Asian music—remains a challenging and underexplored problem, especially in a setting where such genres can closely overlap. In this work, we propose a specialized classification approach leveraging transfer learning with two large pretrained audio models: a transformer-based model and a convolutional architecture. Using small, hand-curated datasets from both American and South Asian musical traditions, we train and fine-tune these models with targeted data augmentations. Our results demonstrate that pretrained networks can be adapted to distinguish subtle differences in related musical styles, despite limited data and overlapping feature spaces. These findings suggest potential for more nuanced genre-aware systems and serve as a foundation for future work in subgenre-specific music generation and interpretability.

1. Introduction

Broad Western music genre classification (rock, pop, rap, etc.) has been performed successfully on several datasets by neural-nets, notably the GTZAN dataset, which is often considered the industry-standard for such tasks by American companies [1, 9, 10, 12]. However, while finer-grain genre classification has been attempted, the results are not nearly as promising, despite findings that deep neural networks can accurately classify scales, notes, keys, and melodies in a variety of musical traditions [3, 4, 6, 7]. We have yet to find a highly-specialized musical model that can perform multilabel classification among a dataset comprising related and sometimes overlapping genres [5]. Additionally, regardless of the state of current research, commercial recommendation systems utilized by music providers like Spotify are noticeably inaccurate at determining much more than such broad and somewhat trivial genre distinctions. The classification of genres and sub-genres of international music seems to be especially weak.

Given these shortcomings, we propose an attempt to build specialized models that are highly accurate at distinguishing between related and overlapping music genres from a diverse set of regions. Building on past successes, we will use transfer learning to finetune existing deep networks that have been pre-trained on large audio datasets. Such finetuning will involve handcrafted, carefully curated and labeled datasets. We will use two large pretrained audio networks, one based on a transformer architecture and the other based on a convolutional neural-net (CNN) architecture. Hyperparameters and architectures of each one will be fine-tuned on two separate datasets, a Western dataset and a South Asian dataset. Preliminary results will compare performance of individual architectures, after which we may broaden the dataset to include a greater diversity of world music, evaluate different architectures for additional applications, or ensemble different architectures together.

Potential challenges of this project are that the model might not generalize well to new data (the networks might not have seen very much music of certain genres in their pre-training). Additionally, genre classification between related genres is inherently a difficult task due to overlaps in instrumentation, rhythm, and style. These challenges are compounded by the fact that we will have small datasets because they must be hand-labelled. However, we believe these challenges can be overcome through careful architecture design and informed data augmentation. If we are able to build models that achieve high accuracy on test data, we hope that these models may serve as foundations for music generation within or at the intersections of specific genres, as well as sources for mechanistic interpretability and human-understandable intuition regarding the features of related and overlapping music genres.

2. Related Work

2.1. American Dataset

- Zhang and Li achieved 96% accuracy on both the GTZAN and Ballroom datasets using ensembles of CNNs. They demonstrated that deep neural networks can learn to distinguish between different genres of music, even if the genres are very similar. Their work also indicates that models generalize better when given multiple feature representations of music. In the case of the ballroom dataset, they showed how models can be trained effectively to categorize related genres when data for each genre is properly annotated [13].

- Li showed that using pre-trained CNNs (even those pretrained on images and not audio) outperformed smaller custom models at genre classification. Using brief mel-spectrograms of 1.875 seconds, the best pretrained model achieved 75% accuracy on the test set. Their dataset consisted of four highly related specific sub-genres of house music [5].

- Quiqiang et. al trained multiple networks on a large general audio dataset. This dataset contained 900,000 examples of music in addition to other sounds such as speech, trains, toothbrushes, etc. Their fine-tuned CNN14 achieved 91.5% accuracy on the GTZAN genre classification task [12].

- Oramas et. al [7] created a multimodal dataset of hundreds of music genres and did multi-label classification. They tested convolutional network architecture parameters to create a network to classify entire albums. Their best design achieved 88% on their accuracy metric [9].

2.2. South Asian Dataset

Current research on South Asian classical music suggests KNNs can differentiate many raags with optimized features. Without constructed features, KNN and SVM methods are able to binarily classify raags given sufficient data, CNNs are able to classify 5 raags effectively given enough data, and when CNNs are combined with LSTMs and fed copious amounts of data, they can correctly classify 30 different raags.

- In 2016, Gulati et al. constructed a new feature known as a time-delayed melody surface, which captures the melodic profile of a soundtrack, and a KNN model using this feature space classified 30 Hindustani raags with 98% accuracy and 40 Carnatic raags with 87% accuracy, showing that with optimized features, even a typical machine learning algorithm is able to classify raags effectively [4].

- In 2021, Joshi, Pareek, and Ambatkar trained models on a binary raag classification task with 341 files in an 80/20 train/test split. They achieved 98% accuracy using a KNN approach and 95% using an SVM approach. Their features were MFCCs, which are a more compressed version of a mel spectrogram, and this shows that even without constructed features, machine learning algorithms can perform well at binary classification [2].

- This team published again in 2023 showing that a CNN can differentiate 5-classes with 85% accuracy using MFCCs and 89% accuracy using mel spectrograms, suggesting that mel spectrograms contain slightly more information and that the task becomes increasingly difficult with larger numbers of raags [3].

- In 2021, Shah, Jagtap, Talekar, and Gawande showed that a combined CNN and LSTM approach could achieve 99.5% accuracy on 30 raags. Notably, they used a regular spectrogram (no mel scaling), 216 frequency bins, and 10 seconds per sample and 5 seconds overlap between 2 samples, and their dataset was around 83,000 images and 116 hours of music. This demonstrates that deep learning models are able to learn features of raags very well given sufficient data [8].

- In 2021, Gong, Chung, and Glass prototyping an auditory spectrogram transformer that could accept mel spectrogram data and classify AudioSet data up with 95% accuracy [11].

3. Methods

3.1. Datasets

Our American dataset consists of 360 songs equally divided into the genres of bluegrass, country, and folk. Bluegrass, country, and folk are genres with lots of overlap, and there is no strict guideline on what differentiates the three genres from each other. Therefore, the songs that were labeled were quintessential examples of their respective genres. It should also be noted that the songs representing folk and country all come from the 1940's or later, as prior to that decade American folk music and American country music were arguably indistinguishable. Below is a brief overview of each of the three genres.

- *Bluegrass*: characterized by a five-piece band consisting of an acoustic guitar, mandolin, banjo, fiddle, and upright bass. The guitar and mandolin are played with flat picks, while the banjo is played using fingerpicks. This genre is characterized by fast tempos, virtuosic solos, and high "lonesome" vocal timbres.

- *American Folk*: characterized by predominantly acoustic instruments, simpler arrangements, and softer vocal timbres. Often contains similar instruments to bluegrass, but lacks the distinctive rhythmic drive and pulse typical of the former. The message of the vocalist is usually the highlight of the song in this genre.
- *Country*: characterized by instruments such as acoustic and electric guitar, drums, harmonica, fiddle, and slide guitar. Has the most blues influence of the three genres and is also the most similar to rock and roll. Country music blends the narrative simplicity and foundation of folk music with the groovy, rhythms and twangy sounds of bluegrass.

Our South Asian dataset consists of 1103 Hindustani classical music soundtracks, divided into 61 Hindustani raags. Raags can be quite similar (for example, Megh Malhar is thought of as a combination of raags Megh and Malhar), and as a result, certain melodies are common between them. Hindustani classical music has been recorded since the technology arrived in British India, and this dataset is representative of soundtracks recorded with varying fidelity and technology, attempting to ensure classification generalizes well across these periods.

We will not attempt to enumerate all 61 raags and their intricacies here, but simply put, a raag is a "type" of scale that is based around a central note, such that the same raag can be played centered around C, D, E, and so on. Raags may have different numbers of notes, as well as different notes played during the ascending and descending scales; however, they are always played in pitch order. The natural, melodic, and harmonic scales in Western music have the same relationship to each other as different raags. Due to the sheer volume of musical traditions in South Asia, hundreds of raags exist in Hindustani music, and hundreds more in Carnatic, and different notes or patterns in the scales are emphasized by these various traditions, further complicating matters.

3.2. Data Pre-Processing and Augmentation

Songs from each American genre were split into a training set, validation set, and test set (split 75/10/15 respectively). Each song was then split randomly into 20 second segments, as this seems. Each segment was then pitch-shifted by a random value between -2 semitones and +2 semitones. Finally, data in each set was randomly shuffled. At train time inputs are converted from their native waveforms into a mel-spectrogram, which is a visually dense representation of the audio. Audio duration is on the x-axis, frequency (pitch) is on the y-axis, and within each frequency bin the loudness of that frequency is indicated by color. Also at train time, within a batch, inputs are augmented using SpecAugment, which randomly masks out horizontal and vertical patches of the mel-spectrogram.

Hindustani classical soundtracks were divided with an 80/10/10 split into training, validation, and test sets, quantified by soundtrack length. Soundtracks were divided into 28,942 30-second segments, shuffled, and converted into a mel-spectrogram capped at 8 kHz and segmented into 128 log frequency bins and 10 ms time bins. Data were augmented by randomly increasing and decreasing the pitches and speeds of the tracks within the range of human perception and enjoyment, as well as SpecAugment.

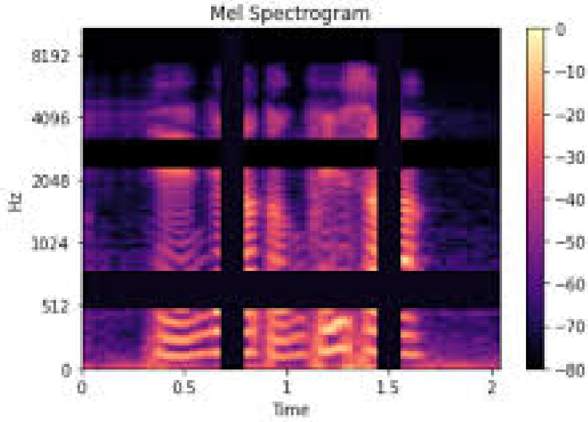


Figure 1. Example of SpecAugment on a mel spectrogram. This example shows 4 total stripe augmentations.

3.3. Model Architectures

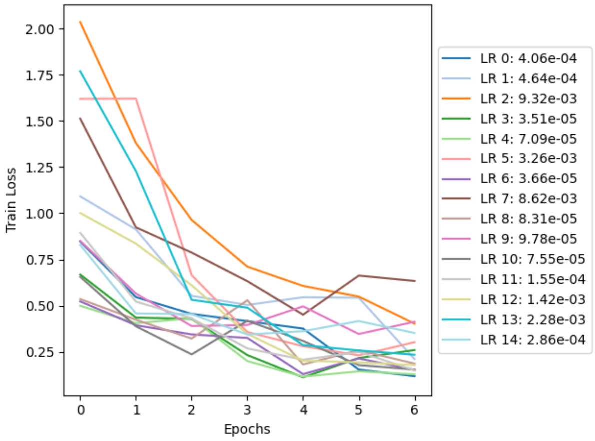
Our first model is built off of the PANN CNN-14 [9] architecture, which consists of six convolutional layers with two intermittent 2x2 pooling layers, then a final global pooling layer followed by two fully connected layers. Our model replaces the final fully connected layer with a fully connected layer of size four for classification. Fine-tuning is then done by freezing all existing weights and only training the newly added 8192 weights. These weights are initialized using PyTorch's default Haiming/Ke initialization.

Our second model is built off of the AST architecture [https://arxiv.org/abs/2104.01778], which is a transformer of mel spectrograms based on a visual transformer. First, the model splits the given mel spectrogram into square patches, of 16 time bins and 16 frequency bins, with an overlap of 6 bins between each square patch and its adjacent patch. It then learns a 768-dimensional embedding for each patch, and uses a 12 layer encoder with 12 self-attention heads per layer to perform classification. A final layer was added with 61 dimensions for the 61 classes of raags in this dataset, with Haiming/Ke initialized weights.

4. Experiments

4.1. CNN14

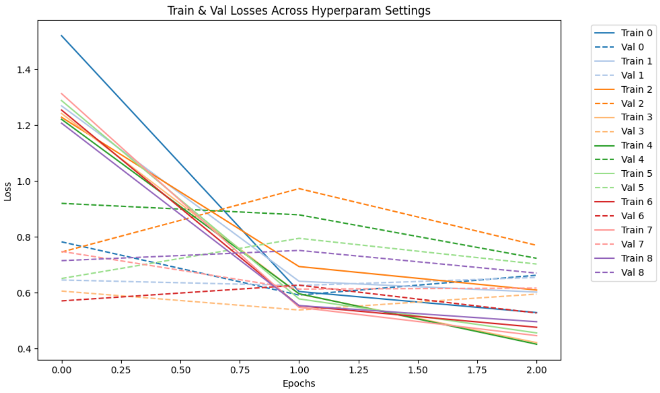
First, a variety of randomly generated learning rates were tested on 20% of the Western training data for 6 epochs. Note that all CNN14 training iterations will be performed with the Adam optimizer using default values of β. The results are shown below:



From this, three candidate learning rates — 9.32e-03, 4.06e-04, and 1.42e-03 — were selected based off of shape of their respective loss curves. These learning rates were then tested on 40% of the training data using a grid search (with slight random perturbation) with L2 regularization values of 1e-5, 5e-5, and 1e-4. The results are shown below:

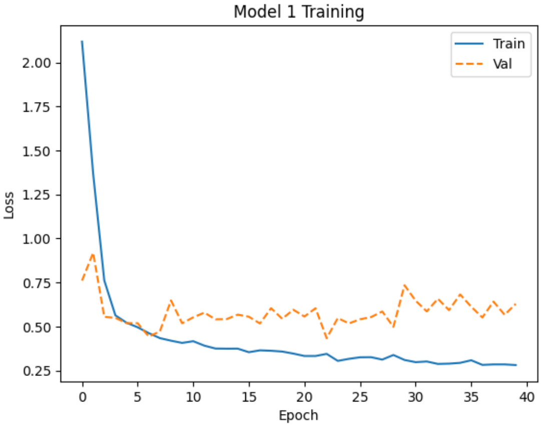
Iter.	LR	Reg	Loss E1	Loss E2	Loss E3
0	9.64e-03	9.56e-06	1.5208	0.6040	0.5285
1	8.89e-03	4.65e-05	1.2695	0.6407	0.6019
2	8.71e-03	9.30e-05	1.2287	0.6934	0.6092
3	3.92e-04	9.95e-06	1.2423	0.5953	0.4207
4	3.71e-04	4.76e-05	1.2213	0.5960	0.4151
5	3.80e-04	1.06e-04	1.2884	0.5774	0.4555
6	1.55e-03	9.12e-06	1.2544	0.5528	0.4757
7	1.56e-03	4.92e-05	1.3134	0.5472	0.4458
8	1.68e-03	1.05e-04	1.2074	0.5535	0.4957

Table 1. Training losses across three epochs for different learning rates and L2 regularization values.

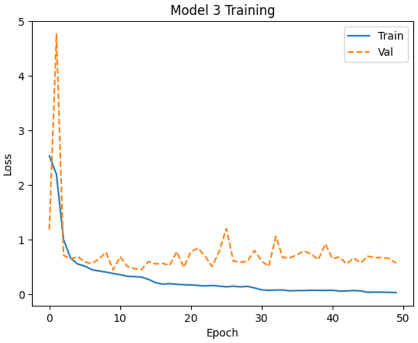


Based on initial tuning, a learning rate of 3.71e-04 and L2 regularization of 8.76e-05 were selected. Subsequent training was performed over many epochs on the full dataset. While the base learning rate remained constant at 3.71e-04, a learning rate scheduler was employed, with both the number of warmup steps and the decay interval (epochs before halving the learning rate) treated as hyperparameters. Other hyperparameters included batch size, weight decay, dropout rate in the final layer, the number of input augmentations, and the number of unfrozen model layers. Below are the results for the training iterations:

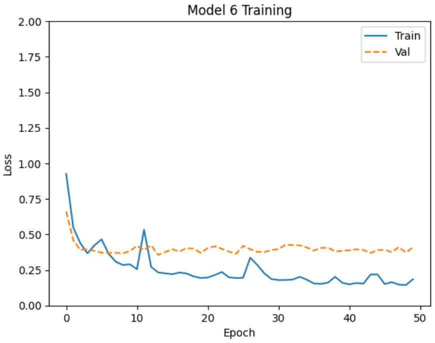
Batch size = 64
Weight regularization = 4.76e-05
Learning rate warmup steps = 0
Decay interval = 10 epochs
Final layer dropout rate = 0.3
Number of augmentation strips = 4
Number of unfrozen final layers = 2



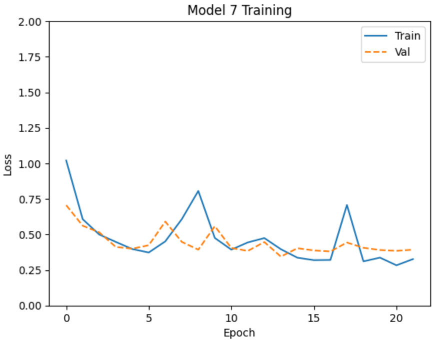
Batch size = 64
Weight regularization = 5.76e-05
Learning rate warmup steps = 250
Decay interval = 15 epochs
Final layer dropout rate = 0.4
Number of augmentation strips = 8
Number of unfrozen final layers = all



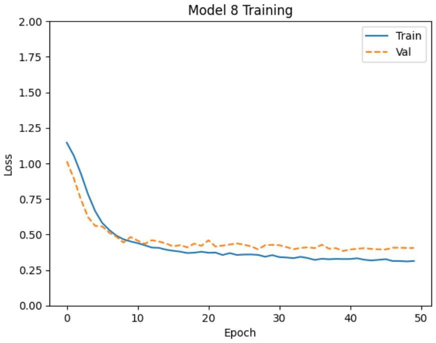
Batch size = 64
Weight regularization = 4.76e-05
Learning rate warmup steps = 250
Decay interval = 10 epochs
Final layer dropout rate = 0.3
Number of augmentation strips = 8
Number of unfrozen final layers = 2



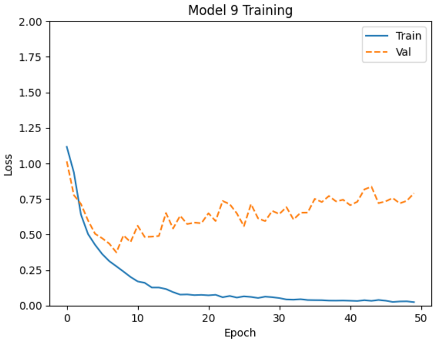
Batch size = 64
Weight regularization = 5.76e-05
Learning rate warmup steps = 500
Decay interval = 15 epochs
Final layer dropout rate = 0.3
Number of augmentation strips = 16
Number of unfrozen final layers = 2



Batch size = 128
Weight regularization = 5.76e-05
Learning rate warmup steps = 500
Decay interval = 15 epochs
Final layer dropout rate = 0.4
Number of augmentation strips = 24
Number of unfrozen final layers = 2

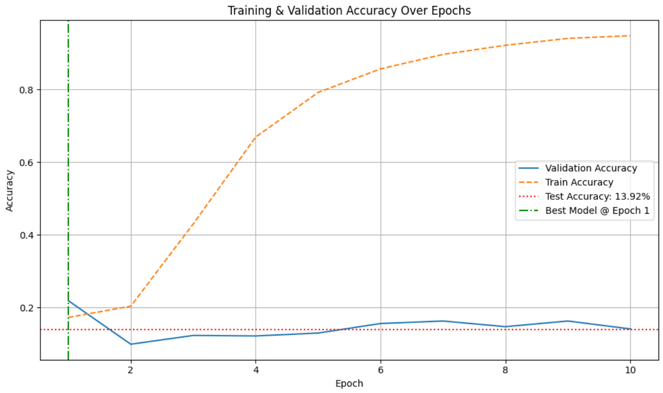


Batch size = 128
Weight regularization = 8.76e-05
Learning rate warmup steps = 500
Decay interval = 15 epochs
Final layer dropout rate = 0.4
Number of augmentation strips = 24
Number of unfrozen final layers = 3

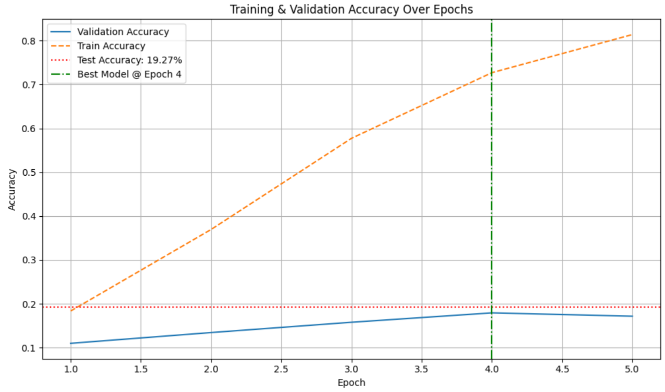


For the South Asian dataset, learning rates were logarithmically grid searched between 1e-6 and 1e-1 (inclusive), and dropout was grid searched between 0.2 and 0.5 (inclusive, increments of 0.1). The number of AST layers frozen was chosen between 0 (control), 8, and 10; freezing all but one layer would likely have resulted in unnecessarily considering features that were irrelevant to music, as AudioSet, which AST was trained on, included non-musical soundtracks. Below are two sample graphs:

Batch size = 8
Dropout rate = 0
Frozen layers = 0

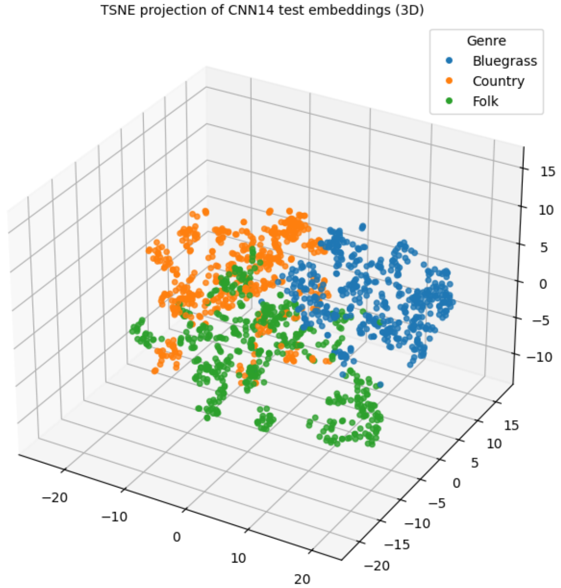


Batch size = 8
Final layer dropout rate = 0.2
Augmentation strips = 4
CutMix probability = 50%
Frozen layers = 8
Weight decay = 1e-4



4.2. Results

Based on these experiments, Model 8 was selected as the final model for the Western dataset. It achieved a raw accuracy of 81% on individual held-out test samples, and its accuracy increased to 83% when predictions were aggregated using a majority vote across entire songs. The following figure shows the results of applying t-SNE (a non-linear dimensionality reduction technique) on the embedding vectors produced by passing the test dataset through the model. Each dot represents a sample in the test dataset, and the separation between genres in the projection demonstrates genre-specific structures learned in the embedding space.



In the case of the South Asian dataset, the model fit the training data well due to its complexity. On validation and test data it performed significantly better than chance (19.23% vs. 1.6%), representing a roughly 12X improvement; however its accuracy was still very low. Given the breadth of hyperparameters searched, we hypothesize that this is likely due to the limitations of the South Asian music dataset, which has roughly 18 examples per raag due to its 61 categories, and is thus prone to overfitting. Additional datasets have recently been made available to us, and we will endeavor to incorporate these into a larger dataset with a smaller number of classes to better evaluate our hypothesis before extrapolating to all 61 classes.

5. Discussion

While the task was successful with a simpler model for a small number of genres on a larger dataset, a more complex model, a smaller dataset, and a larger number of classes resulted in overcomplexity for an ill-defined task.

- Our current network for South Asian music vastly overfits the training data; continued work will be focused on remedying this. https://colab.research.google.com/drive/1AwiBNa0doSKZPw00io2hTtpLnMWygbdw?usp=sharing

- Data extraction pipeline for Western dataset: https://colab.research.google.com/drive/1ntc1qDcIBs-vjPKN2mHPfZdtgv8eearr?usp=sharing

- CNN14 Training and Evaluation: https://colab.research.google.com/drive/

17NgYBDo_k1qRXZT_c-0Tnt_NtgjeS8J3?usp=sharing

Future directions of the project include:

- Interpret meaning of clusters of American genres into meaningful musical characteristics
- Improve size of dataset for raaga classification
- Train raaga classification on less complex models to encourage learning
- Train and evaluate performance of models pretrained on each dataset on the opposite dataset
- Interpret feature spaces that separate clusters within and between datasets
- Train and evaluate combinations of the two model architectures on each dataset
- Interpret intermediate neurons and features within the models

References

[1] Mudassir Imam Juan Francisco Leonhardt and Yu Wang. Music genre classification: A machine learning exercise. Medium.com, May 10, 2024. Accessed: April 30, 2025. Available: https://medium.com/@juanfraleonhardt/music-genre-classification-a-machine-learning-exercise-9c83108fd2bb. 1

[2] Dipti Joshi Jyoti Pareek and Pushkar Ambatkar. Indian classical raga identification using machine learning. In Sarika Jain and Sven Groppe, editors, *Proceedings of the International Semantic Intelligence Conference*, volume 2786, pages 259–263, New Delhi, India, 2021. 2

[3] Dipti Joshi Jyoti Pareek and Pushkar Ambatkar. Comparative study of mfcc and mel spectrogram for raga classification using cnn. *Indian Journal of Science and Technology*, 16:816–822, 2023. 1, 2

[4] Sankalp Gulati Joan Serrà Julià Kaustuv Kanti Ganguli, Sertan Sentürk and Xavier Serra. Time-delayed melody surfaces for rāga recognition. In *17th Int. Soc. Music Inf. Retrieval Conf.*, pages 751–7, 2016. 1, 2

[5] Xinyu Li. Housex: A fine-grained house music dataset and its potential in the music industry. *arXiv*, 2022. 1, 2

[6] Cheng-Zhi Anna Huang Ashish Vaswani Jakob Uszkoreit Noam Shazeer Ian Simon Curtis Hawthorne Andrew M. Dai Matthew D. Hoffman, Monica Dinculescu and Douglas Eck. Music transformer. *arXiv*, 2018. 1

[7] Priya Mishra and Manish Kalra. Transformer-based technique to classify raags in hindustani classical music. IEEE Conf. Interdisc. Appr. Tech. Man. Soc. Innov., 2022. 1

[8] Devansh P Shah Nikhil M Jagtap, Prathmesh T Talekar and Kiran Gawande. Raga recognition in indian classical music using deep learning. In *Artificial Intelligence in Music, Sound, Art and Design*, volume 12693, Seville, Spain, 2021. EvoMUSART, Springer. 2

[9] Sergio Oramas Oriol Nieto, Francesco Barbieri and Xavier Serra. Multi-label music genre classification from audio, text, and images using deep features. *arXiv*, 2017. 1, 2

[10] Swati A Patil Pradeepini Gera and Thirupathi Rao Komati. Novel mathematical model for the classification of music and rhythmic genre using deep neural network. *Journal of Big Data*, 10(108), 2023. 1

[11] Yuan Gong Yu-An Chung and James Glass. Audio spectrogram transformer. *arXiv*, 2021. 2

[12] Qiuqiang Kong Yin Cao Turab Iqbal Yuxuan Wang, Wenwu Wang and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *arXiv*, 2020. 1, 2

[13] Yuxin Zhang and Teng Li. Music genre classification with parallel convolutional neural networks and capuchin search algorithm. *Scientific Reports*, 15(9580), 2025. 2