Music Genre Classification Mid Project Report

Yuhan Wei
yw157@rice.edu

Xiaoyu Chen
xc55@rice.edu

Yiwei Lu
yl281@rice.edu

1 Problem Description

Music genre classification is an important step of music search and recommendation, but it suffers with the following three main problems:

1. Misclassification, unlike the other data annotation tasks, music genre classification is more challenging. The standard practice of classifying music genres based on the singer or the music publisher can result in low accuracy of music labeling, which will negatively impact the user experience.
2. Data process may degrade the performances by losing significant features. Excessive features will entail heavy computation and slow the system's efficiency. How to extract enough significant features efficiently is the key problem.
3. Limited memory and low-power hardware. The platform would have to be able to run the model on hardware with modest memory and power requirements, like mobile phones. Since device memory is limited, an efficient model must be created.

Our proposal for addressing the three problems are as follows:

1. To improve the accuracy of music classification, we have developed a multi-modal neural network that combines a convolutional neural network (ResNet18) and a recurrent neural network (LSTM). This model has been trained using both images (such as spectrograms) and time series data (such as MFCCs) extracted from audio segments. By utilizing both modalities, we have achieved a significant improvement in music classification accuracy.
2. To capture the audio signal sensitively and effectively, we have divided each signal into different scaling small segments and extracted features to obtain MFCCs(Mel-frequency cepstral coefficients). Using graphics, and amplitude are worked as the time domain does not sufficiently classify music genre, So the Fourier transform is introduced as the frequency domain. We have developed a convolutional neural network(CNN) to test the raw data and processed data, the raw test dataset accuracy is almost half that of the processed test dataset.
3. To create an efficient model, first we will utilize advanced deep learning models. Specifically, we will use ResNet18 for image-based representation and MLP with dropout layers for numerical music features. Then we will apply efficient model compression techniques such as pruning, quantization, and knowledge distillation. These methods aim to reduce the size of the models without significantly affecting their performance, making them suitable for deployment on devices with limited memory.

2 Literature Survey

2.1 User Case 1: Multi-modal Neural Network

Machine learning and deep learning techniques are widely used in the area of music information retrieval (MIR). Hareesh Bahuleyan [1] proposed two different approaches to classify music, one is VGG-16 fine-tuned on spectrogram and the other is XGBoost trained with important music features. Their results indicates that the ensemble classifier of VGG-16 and XGB proved to be the most beneficial and achieved 65% accuracy. S Oramas et al. [3] proposed a representation learning approach for the classification of music genres from different data modalities, i.e., audio, text, and images. The results show in both scenarios that the combination of learned data representations from different modalities yields better results than any of the modalities in isolation.

The present study leveraged existing machine learning algorithms and pre-trained CNN models to enhance the model's performance. We propose a similar multi-modal approach that combines CNN and RNN to achieve better prediction accuracy.

2.2 User Case 2: Data Processing

Data processing is to convert raw data into an information form and provide diverse models to make decisions or automate processes.

Audio feature extraction has multiple ways. The multi-resolution analysis in [5], utilized time-frequency features via MFCCs, visualization of frequency on time. The article mixes features by the audio signal and time series, but the audio signal could induce redundant data which could heavier the computational cost. Some similarities between two signals will result in dependency linearly, which may cause incorrect relationships and errors.

CNN, one of the popular speech recognition models [6] could be introduced for music classification. By converting the MFCCs into images with fixed width and length and creating tuples, this preprocessing audio methodology has achieved 0.5196 accuracies.

2.3 User Case 3: Model Compression

Model compression has been an active area of research in recent years, with the growing demand for deploying deep learning models on resource-constrained devices such as mobile phones and IoT devices.

Pruning, a technique for model compression that has received comprehensive investigation, attempts to remove redundant or useless weights or neurons from the network while maintaining performance. According to a method named "Deep Compression" brought out by Han et al. (2015)[4], large compression rates can be attained without a major loss in accuracy by combining pruning, weight sharing, and Huffman coding.

Quantization is another popular model compression method that reduces the number of bits used to represent weights and activations in the network. Courbariaux et al. (2015)[2] proposed the binary weight network (BWN), where weights are quantized to either -1 or 1, achieving considerable model size reduction.

3 Dataset

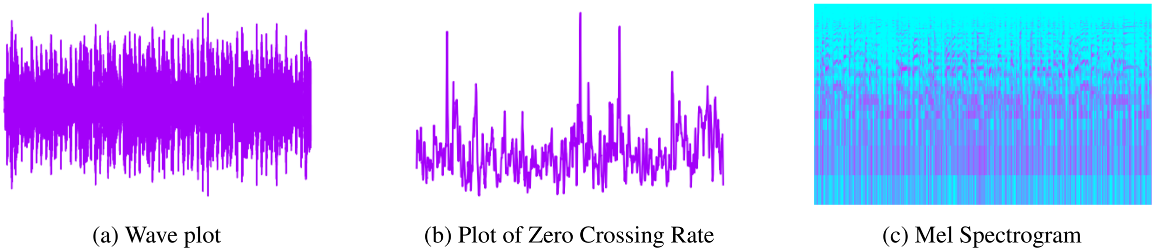
The raw audio data is from GTZAN Genre Classification dataset, which is a widely-used dataset in the field of machine learning and audio analysis. The dataset consists of 1,000 audio files, each 30 seconds long, with 10 different genres of music: blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock. The audio files are in WAV format and have a sampling rate of 44.1 kHz with 16 bits per sample. GTZAN dataset is commonly used for evaluating the performance of machine learning models in music genre classification tasks.

4 Multi-modal Neural Network

We hypothesize that utilizing multiple modalities of data such as visual (image-based) and auditory (sound-based) features can improve the accuracy of music genre classification compared to using only one modality, which is significant for building an accurate music searching and recommendation system.

4.1 Data

Initially, the raw audio data is transformed into several images that showcase different audio features, including wave plots and Mel spectrograms. As demonstrated in Figure 1, an audio file belonging to the blues genre was utilized as an example, where the features like waveforms and zero crossing rate were extracted from the music segments, and then converted into image forms for further classification.



(a) Wave plot

(b) Plot of Zero Crossing Rate

(c) Mel Spectrogram

Figure 1: Three different visual representation of a blues music

To further enhance our dataset, we extracted Mel-frequency cepstral coefficients (MFCCs) from each audio segment based on the time horizon. This enabled us to create a time series dataset, where each sample was comprised of 1290 time steps. The feature of each time step was represented as a vector of length 13, resulting in a final dataset with samples of dimension 1290 × 13.

4.2 Method

We proposed a multi-modal neural network (figure 2) for music genre classification that combines a convolutional neural network (CNN) and a recurrent neural network (RNN). The CNN was used to process visual (image-based) features, while the RNN was used to process auditory (time series-based) features. The output vectors from the CNN and RNN were concatenated and passed through several fully-connected layers to generate the music genre prediction.

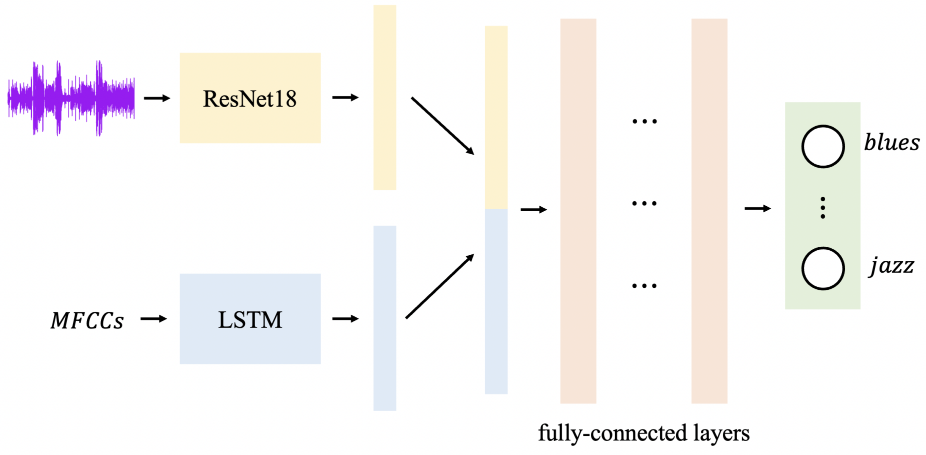


Figure 2: Multi-modal neural network architecture

The CNN component of the network utilized the ResNet18 architecture, which has been shown to perform well on image classification tasks. We fine-tuned this pre-trained model on our dataset of music images, allowing the network to learn higher-level features specific to the music genre classification task.

The RNN component of the network utilized a long short-term memory (LSTM) architecture, which is well-suited to modeling sequential time series data such as audio signals. We trained the LSTM on our dataset of MFCCs, allowing the network to learn temporal patterns and dependencies in the data.

We split the data into 90% training and 10% validation sets using stratified shuffle to ensure that each class was proportionally represented in both sets, and compared our model's performance to baseline models that used only one modality of data (i.e., either images or time series). The evaluation metrics we used in our experiments was prediction accuracy.

4.3 Experiments and Results

We conducted our experiments using the deep learning library *PyTorch* on Google Colab to accelerate the process. The baseline models were trained using either images or time series to predict music genres. The prediction accuracy for all baseline models is shown in Table 1. The ResNet18 model trained with wave plots or Zero Crossing Rate plots performed poorly, with the best model achieving only around 50% accuracy. The LSTM model trained with sequential MFCCs achieved only about 40% accuracy. However, the ResNet18 model trained with Mel Spectrograms achieved over 80% accuracy due to the additional information contained in the spectrograms. Our further experiments will focus on whether the multi-modal network outperforms the baseline models by utilizing the multi-modal information.

5 Data Processing

We hypothesize that time-domain and frequency domain obtaining mel frequency cepstral coefficients can capture enough signal representation on music classification through segment analysis on CNN by using less computational time and cost.

Table 1: Validation accuracy for baseline models

Modality	Dataset	Model	Accuracy
Images	Wave plot	ResNet18	0.48
	Plot of Zero Crossing Rate	ResNet18	0.50
	Mel Spectrogram	ResNet18	0.84
	Plot of MFCCs	ResNet18	0.70
Time Series	MFCCs	LSTM	0.42

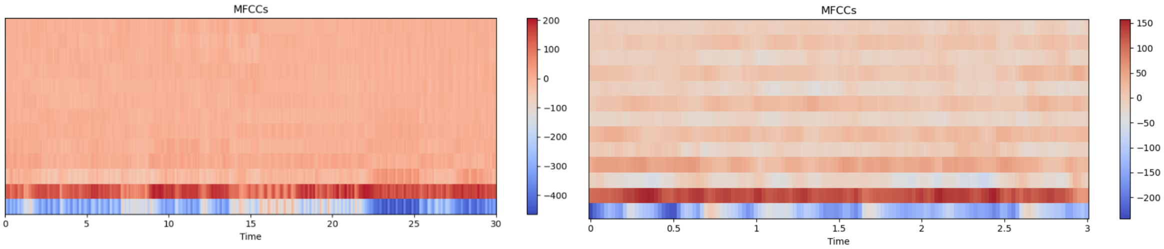
5.1 Data

Before processing the data, various methods of visualization of the audio files are significant. GTZAN dataset has ten music genres and sub-music samples that last for 30 seconds, which could be processed into data segments. After loading the raw audio files in the dataset, we split them into three seconds segments and calculated the MFCCs for each piece. When calculating the MFCCs, graphics are used to order the time frequency, broadening the informative presentations. According to the genre names, we used LabelEncoder to label each audio file and divide the audio files into classes.

Based on the shape of MFCCs, we build the CNN model, some convolution layers, and pooling layers and obtain the parameters for the last layer. Divide the dataset into 80% on trainset, and 20% on testset.

5.2 Method

We use AudioSegment from pydub package to divide the single audio file into 10 pieces that each last for 3 seconds. Later we may vary the length of each piece, and test the information obtained from each piece. Choose the optimal time length of each piece to get the best performance of music classification. Then used feature from librosa package to extract MFCCs for a single piece. To get rid of any type of noise, we will use discrete cosine transform (DCT) to discard the noise. The MFCCs could not be able to provide information on the loudness of the audio segment. So Root Mean Square(RMS) is introduced as a tool to measure the soundness of each frame. We train 9 layers of the CNN model and compare the length of each frame that concentrates various pieces of features together. In comparison with various genres, the accuracy for each genre is slightly different. Some comparison and analysis for audio segments could be revealed by figuring out compute pairwise distances on Euclidean distance and correlation distance metrics.



(a) MFCCs for three seconds window

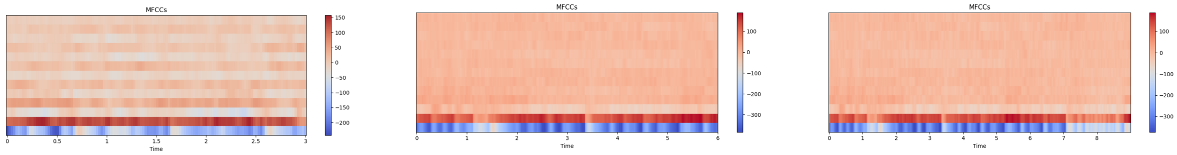
(b) MFCCs for single window

Figure 3: Mel-frequency cepstrum coefficients on raw data Jazz music

5.3 Experiments and Results

The visualization in figure 3 of Jazz raw audio files after trimming and one frame of 3 seconds file could indicate fewer features involved in a single frame. The length for single frame increases accordingly as time periods varies. The 9 layers of CNN trained the model with the batch size = 16, epoch = 40, generating around 95% accuracy. Default number of MFCCs to return is 13 based on the input size for CNN. Since the shape of frame for

6 seconds and 9 seconds is different than the 3 seconds, the input layer parameters of CNN varies. The number of CNN parameters$(2, 855, 818)$ for 3 seconds frame is the baseline.



(a) MFCCs for three seconds segment (b) MFCCs for six seconds segment (c) MFCCs for nine seconds segment

Figure 4: Different time length for a single frame of raw data Jazz music

6 Model Compression

We hypothesize that efficient model compression techniques can significantly reduce deep learning model sizes without compromising their performance, making them suitable for deployment on resource-constrained platforms such as mobile streaming devices.

6.1 Data

For preparing the image data, we use the original images as the base data. We first set the batch size to 8 and the image size to 224 for the data processing pipeline. By applying separate transformation pipelines for training and validation datasets, and loads the data using torchvision's ImageFolder. It encodes and decodes class labels, sets a random seed for reproducibility, splits the data into training and validation sets, and creates data loaders with specific configurations to optimize GPU processing.

For the music feature model, we use the existing file as input, spliting it with the following distribution: 70% on training set, 20% on dev set, and 10% on test set.

6.2 Method

We compress the image model and music features model to make them fit for platforms such as mobile streaming. For the image model, ResNet18 is used as the base architecture, and we apply a combination of techniques for compression, such as pruning, quantization, and knowledge distillation. We create a custom quantized ResNet18 model and update the net class definition, which is the customized model designed for the original ResNet18 model, to use the custom quantized ResNet18. The torch library is used for preparing model quantization.

For the music feature-based model, a Multi-Layer Perceptron (MLP) with dropout layers is employed as the training model. To compress the MLP model, we use the model pruning technique provided by the TensorFlow Model Optimization Toolkit, which removes unnecessary weights and connections, reducing the model size while preserving its performance.

6.3 Experiments and Results

To evaluate the performance of the quantized model, we record the accuracy and average inference time for the original model and the quantized model, and we calculate the compression ratio. The original model achieves an accuracy of 0.7374 with an average inference time of 0.14883461 seconds, while the quantized model reaches an accuracy of 0.7273 with an average inference time of 0.14791101 seconds. The compression ratio for the quantized model is 3.93.

For the music feature-based model, both the original and pruned models achieve an accuracy of 0.9274, with a compression ratio of 5.49 for the pruned model. The effectiveness of the pruning procedure can be attributed to the model's redundant or non-critical weights being removed without significantly impacting performance.

References

[1] Hareesh Bahuleyan. Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*, 2018.

[2] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *arXiv preprint arXiv:1511.00363*, 2015.

[3] Sergio Oramas, Francesco Barbieri, Oriol Nieto Caballero, and Xavier Serra. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*, 2018.

[4] William J. Dally Song Han, Huizi Mao. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. 2015.

[5] Sergey Voronin and Alexander Grushin. A multi-resolution approach for audio classification. 2018.

[6] Dwi Sari Widyowaty, Andi Sunyoto, and Hanif Al Fatta. Accent recognition using mel-frequency cepstral coefficients and convolutional neural network. In *International Conference on Innovation in Science and Technology (ICIST 2020)*, pages 43–46. Atlantis Press, 2021.