Music Genre Classification Using Deep Learning Techniques

Peter Schaldenbrand
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pschalde@cs.cmu.edu

Yizhou He
McWilliams Center for Cosmology
Carnegie Mellon University
Pittsburgh, PA 15213
yhe2@andrew.cmu.edu

Abstract

Classifying the genre of music using only the music data is a difficult problem. Amongst the reasons for this difficulty are high dimensionality of data, ambiguity of genres, and access to large amounts of training data. To overcome these obstacles prior work [1, 2, 3, 4, 5] has used deep learning models that utilize one dimensional convolutional layers. In this paper we reproduced two models which claim state of the art results [4, 3]. We then improved upon these models by replacing the 1D convolutional layers with 2D convolutional layers. This architecture change allowed the model to generalize better leading to a 5% increase in test accuracy over the next best model.

1 Introduction

1.1 Music Classification

Our goal in this paper was to classify a song's genre using various methods and types of data associated with music. Music classification tasks are notoriously difficult to perform due to the high dimensionality of music. While an image may be of size 100×100, a full length song could be of size $2 \times 8,000,000$ depending on the length of the song and the sample rate it was recorded at. With such large dimensions, it is important to both find models that can represent such complex relationships in data and research alternative ways of representing the data. We chose to use deep learning methods when working with the raw music audio in this paper since the data is so complex. When trying simpler models such as logistic regression, we were unable to learn much about the data.

Another aspect of music that makes tasks such as genre classification so difficult is the fuzzy boundary between genres. Music is an incredibly ambiguous medium that is often up to the interpretation of the listener. We are relying on robust labels of genre for our data, but we recognize that labels are often disputed in music communities and that not even the most musically inclined human could correctly classify every song into a particular genre.

Raw music data is stored in a vector of intensities. These intensities, when played through a speaker, make sense to humans who can listen to the vibrations. However, this vector is not intelligible to people even when visualized. There are multiple ways to extract features that are more intelligible from music, and the most common way is to transform this vector into a matrix of frequency and time. By taking the Fourier transformation of the intensity vector of music, the product will be a matrix with the frequency/pitch on one axis and time on the other. Each value of the matrix is the magnitude with which the pitch was played at a time stamp. These matrices are called spectrograms, and examples of them can be seen in Figure 1. While determining the genre from these spectrograms

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

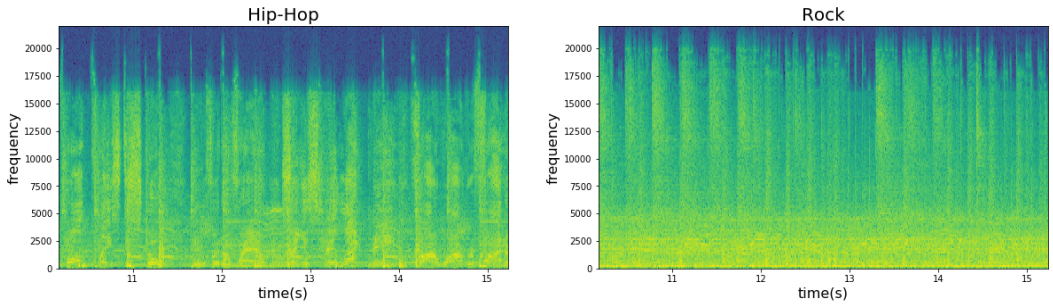

Figure 1: Spectrograms of 2 songs classified to 2 genres Hip-Hop and Rock respectively

may still be difficult to distinguish with the human eye, it is at least far easier to understand than the vector of intensities.

There has been a great number of prior deep learning models that have used spectrograms as input data when classifying music genre. Most of these models [1, 2, 3, 4, 5] use 1D convolutional layers. The 1D layers have filters that operate over the entire frequency dimension of a song, then move through time. The frequency dimension is of size 128 in spectrograms and so the filters are of size 128×3. Models that use 1D convolutions have no trouble minimizing the training loss but have difficulty generalizing to the validation and test sets.

The models that are using 1D convolutional layers are over-fitting, so in order to simplify the model, 2D convolutional layers can be used to reduce the number of parameters. Rather than using filters of size 128×3, filters of size 16×3 can be used in a non-overlapping fashion as seen in Figure 2. Reducing the number of parameters can make the model less complex and generalize better to non-training data. In this paper, we reproduced two high performing genre classification models. We then experimented with replacing the 1D convolutional layers with 2D convolutional layers to observe better generalization of the model.

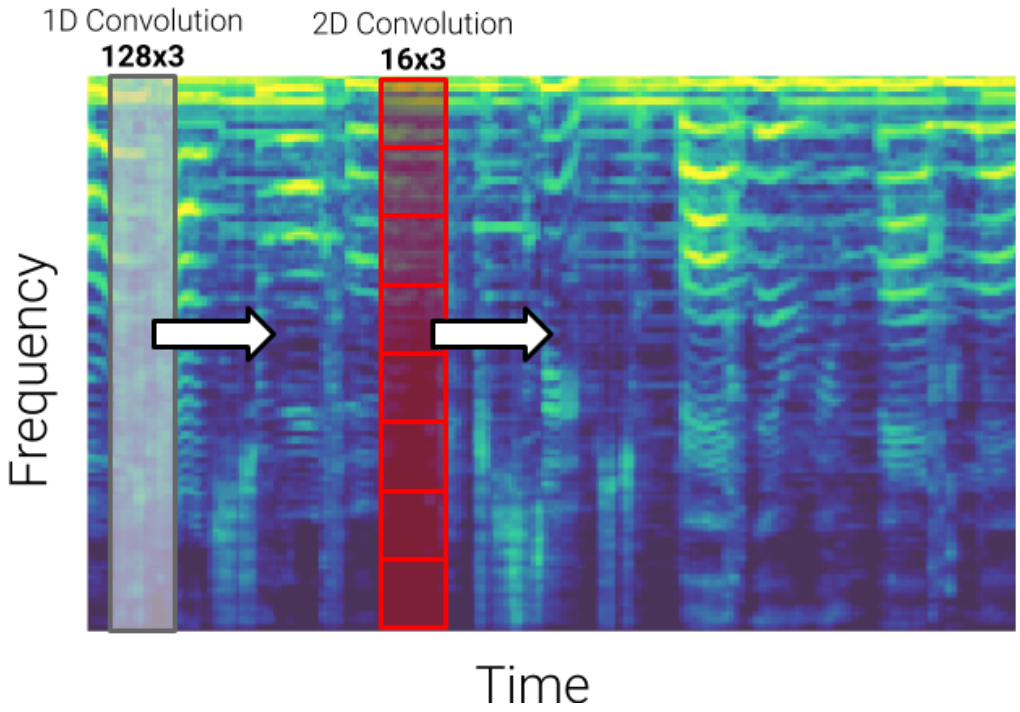

Figure 2: Displaying the differences between using a 1D convolutional filter versus a 2D convolutional filter over a spectrogram.

1.2 FMA Music Dataset

In this paper we used the Free Music Archive (FMA) [6] which is an open and easily accessible dataset that contains both raw music files in the .mp3 encoding and also a large set of meta data associated with each of these files. The dataset contains 917 GB and 343 days of Creative Commons-licensed audio from 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres. In this mid-way report paper, however, we used a subset of the dataset that included 8,000 songs taken equally from 8 different genres.

Each of the songs was converted into mel-frequency cepstrum (spectrograms) using the librosa python package [7]. The number of mel bins was set to 128 as is common in MIR tasks. All songs in the FMA small dataset are approximately 30 seconds long. This translates to a little over 1200 values of mel-frequencies. To ensure that the sizes of all samples for our model were consistent, the time dimension was slightly truncated to 1200 forcing the spectrogram dimensions to be necessarily 1200×128.

2 Related Work

Recent advances in deep neural network training provide an efficient way to recognize patterns in raw data leading to progress in Music Information Retrieval (MIR) tasks. [8] trained a MLP with different activation functions, ReLU and sigmoid, on each neuron and applied different optimizations such as SGD and Hessian Free. They finally reached a 73.5% accuracy on ISMIR 2004 dataset and a 83.0% accuracy on GTZAN dataset [9] for genre classification. The GTZAN dataset used to be a benchmark for genre classification tasks. Different state of art systems has been applied to GTZAN, and amazing results has been reported. For instance, [10] proclaimed an accuracy of 78.0%. They extracted features from spectrograms using a deep convolutional network trained for image classification, and finally used a SVM for genre prediction. [11] reported scores of 87.9% using rich statistics and low-level music features. [12] used a transfer learning system trained for music tagging to extract features for genre prediction. They reported scores of 89.8% by taking features from multiple layers of the transfer CNN model. [13] reported an accuracy of 90.79% using a SVM classifier over selected combination of high and low level musical features. [14] used rich, psycho-physiologically inspired properties of temporal modulations of music with a sparse representation based classifier to achieve accuracy score of 91.0%. Mostly pitch, temporal and timbre features were used with non negative matrix factorization as a feature reduction technique. The same author of [15] further increases the score to 93.7% by the utilization of topology preserving non-negative tensor factorization. More exciting work has been done by [3] proposing a novel approach using an ensemble of convolutional long short term memory based neural networks (CNN LSTM) and a transfer learning model. Their model pushed the accuracy of genre classification on GTZAN dataset to 94.2%. Though genre classification accuracy reported above looks pretty high, it's not difficult to point out that the ISMIR 2004 dataset consisting of 1458 samples over 6 genres and the GTZAN dataset consisting of 1000 sample over 10 genres are still too small to ensure that the models proposed really meet the requirement of MIR tasks of music data in our real life which are way more huge and complex than the data used for training the model mentioned above.

Ambitious scientists then turn their sights to a larger, more representative and reliable dataset FMA. There has been a substantial amount of prior work using the FMA dataset for genre classification. The FMA paper[6] reports results from using various feature engineering packages that take raw audio and output a small feature vector, then using this feature vector as input for various machine learning models for genre classification. For example, they used the MFCC feature extractor [16] with a multi-layer perceptron model to achieve a 53% test accuracy. [17] used full size of FMA dataset with 18045 songs divided into 11 genres and propose a resampling method to balance the dataset, they reported 51.8% weighted accuracy with MLP classifier.

In addition to using the raw audio files with a deep learning feature extractor, other works have attempted to transform music data into more interpretable forms of data. Music is often thought of in terms of pitch (frequency) and time. Using a Fourier transformation on raw audio data will output a matrix whose dimensions are frequency vs. time which is referred to as a spectrogram. In [2], they use spectrograms as input to various machine learning models. Their best accuracy was 51.8% using a multi-layer perceptron network. State-of-the-art models now appear to be using spectrograms as input to complex deep learning models that include convolutional layers in deep belief networks [1].

Recent advancements in recurrent neural network (RNN) research have demonstrated the superiority of utilizing multiscale structures in learning temporal representations of time series. [18] applied LSTM and GRU integrated with a dynamical scaling algorithm to a small subset of FMA data with 8000 songs labelled into 8 balanced genres and achieve an accuracy of 42.4%.

Not confined by a single model classifier, in [19], a two phase classifier is proposed to overcome the problem of blurry classification of fuzzy and distinct genres and report accuracy of 69.9% over 19503 songs of FMA dataset. There is another paper [20] used the kapre audio processing library for the feature extraction, then fed this feature vector into a hierarchical ensemble classifier to achieve a 75.0% accuracy on the test set.

3 Methods

3.1 Methods: Reproducing

In order to test if our 2D convolutional layer would improve a model using 1D convolutional layers, we needed to recreate some of the state of the art music genre classification models. The first model we reproduced was the model in [4, 21], which we will refer to as the Dieleman model.

The Dieleman model is a deep learning network that utilizes 1D convolutions, 1D max pooling, global max and average pooling, and dense layers. The first part of the model is three 1D convolutional layers each separated by 1D max pooling layers. The 1D convolutions have a filter width of 3. This means that the filter is of size 128×3 for the first convolution. The filter operates over the entire frequency dimension and only 3 values of the time dimension. The 1D max pooling layers have a pool size of 4 or 2. This reduces the time dimension by a factor of 4 or 2 but leaves the frequency dimension unchanged. The convolutional layers are followed by a concatenation of global max and average pooling layers. The output of the global max and average pooling layers is connected to two dense layers which have a final size of 8 which corresponds to the number of unique genres. The validation set was used to decide the best number of filters for each convolutional layer and the optimal number of hidden units in the dense layers.

The second paper that we reproduced was [3] which contained two models. The first was similar to the Dieleman model and we will refer to it as the Ghosal model. This model differed from the Dieleman model in that it only had two 1D convolutional layers. After the convolutional layers was a global max pooling layer and a dense layer to get the output of size 8.

The second model from [3] was the same as the first in the paper but they replaced the global max pooling layer with an LSTM [22] layer. This model was very similar to the model in Figure 3 but with 1D convolutions instead of 2D. We refer to this model as the Ghosal LSTM model. The LSTM layer only outputted the last hidden layer which was then reduced to 8 in size by a dense layer.

3.2 Methods: 2D Convolution

We recreated the results of the three models mentioned above (Dieleman, Ghosal, Ghosal LSTM) then determined which model was the best performing be measuring the validation loss. The validation set was a hold out set not used for training. The best performing model was the model that minimized the test loss.

The Ghosal LSTM model had the lowest test loss so we chose to alter this model in order to improve it. As seen in Figure 2, a 1D convolution can be broken up with small filters in a 2D convolution. The Ghosal LSTM model uses a 128×3 size filter in its first convolutional layer. This leads to having 49280 trainable parameters in this layer since there are 128 filters. With the 2D convolutional layer, the filter is of size 16×3. This filter is used in a non-overlapping way over the frequency dimension, so it essentially reduces the frequency from size 128 to 8. Because the filter size is smaller, the number of trainable parameters is greatly reduced in the 2D convolutional layer to 6272. The visual details of this model are summarized in Figure 3.

In order to determine whether the increase in accuracy was caused by the 2D convolutional layer or the LSTM, we also implemented the Ghosal model without the LSTM layer and used 2D convolutions rather than 1D.

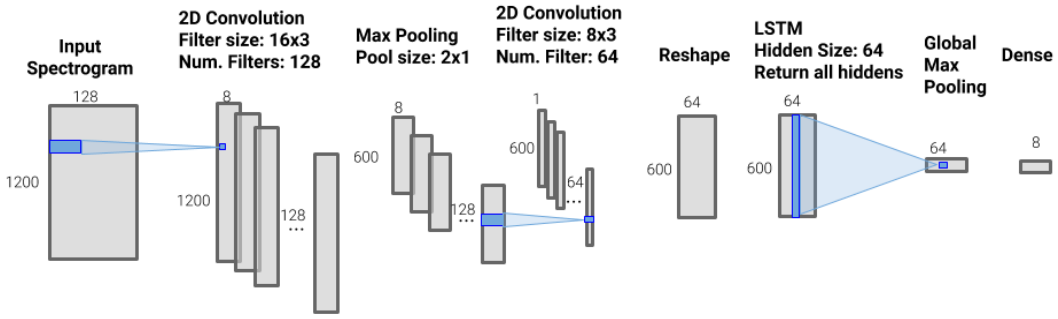

Figure 3: Architecture of the extension of the Ghosal LSTM model [3] to use 2D convolutions instead of 1D.

Model	Test Accuracy	Test Loss
Dieleman [4]	45.63%	1.517
Ghosal [3]	48.88%	1.462
Ghosal LSTM [3]	49.50%	1.460
2D Conv. Model	53.38%	1.403
2D Conv. + LSTM Model	54.75%	1.364

Table 1: Results of the various models in this paper tested on 800 holdout test cases from the FMA dataset.

4 Results

The four models outlined in the methods section were trained on 80% of the FMA dataset which equated to 6329 cases. The models were validated on a separate 10% of the dataset. Each model was trained using the Adam optimization algorithm [23] and utilized categorical cross-entropy loss. The validation loss during training is summarized in Figure 4.

Each model was trained until the validation loss did not improve for 25 epochs. The model that minimized the validation loss was then saved. Using these saved models, each was evaluated on the test set which was 10% (800 cases) from the FMA dataset that were held out and not used for any training, validation, or parameter searching. The results are summarized in Table 1. All of the models were able to over-fit. The training loss is summarized in Figure 5 and shows that all of the models are complex enough to achieve extremely low loss on the training set. But as training loss continued to decrease, validation loss would eventually increase.

The test results of the Dieleman, Ghosal, and Ghosal LSTM models were determined, then used to decide which to update with 2D convolutions. Since the Ghosal LSTM model minimized the test loss, this model was modified to use 2D convolutions instead of 1D convolutions.

Switching to the 2D convolutions greatly improved the Ghosal LSTM model both in validation loss and test loss. The test accuracy rose by more than 5% with this switch compared to the 1D convolution model.

We wanted to make sure that it was the 2D convolutions that were improving Ghosal's model and not the LSTM, so we also tested the 2D convolutions in the Ghosal model that did not have an LSTM. This modified model had a test accuracy of 53.38%. This is larger than the 1D convolution models but less than the 2D convolution and LSTM model.

5 Discussion

The Dieleman model [4] was used to classify genre on Spotify's music data. Spotify is a music streaming company that has access to most recorded music. The Dieleman model detailed in [21] contains 2048 hidden units in the dense layers and 256 to 512 filters in the convolutional layers. We tested the model using these parameters on the FMA dataset and they caused the model to over-fit

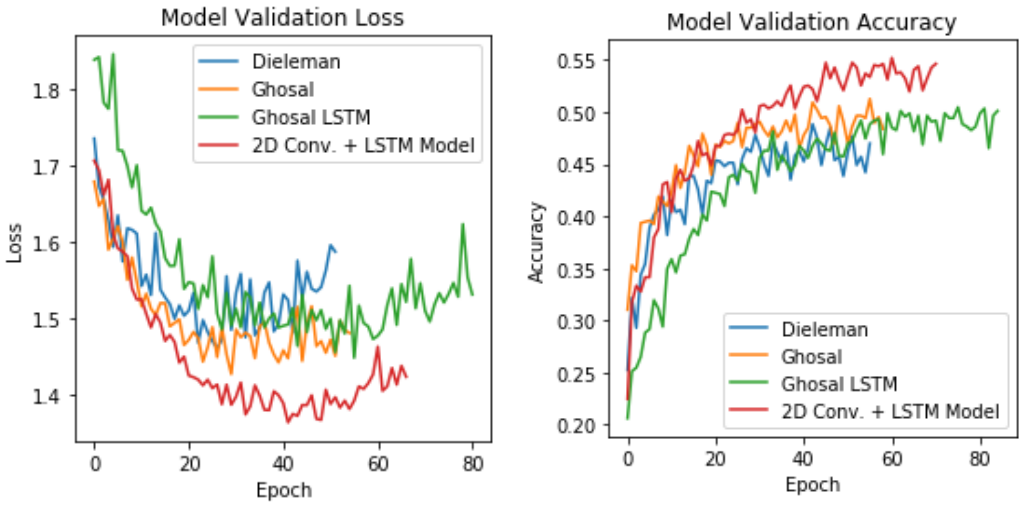

Figure 4: Validation loss and classification accuracy of the four models in this paper over the epochs.

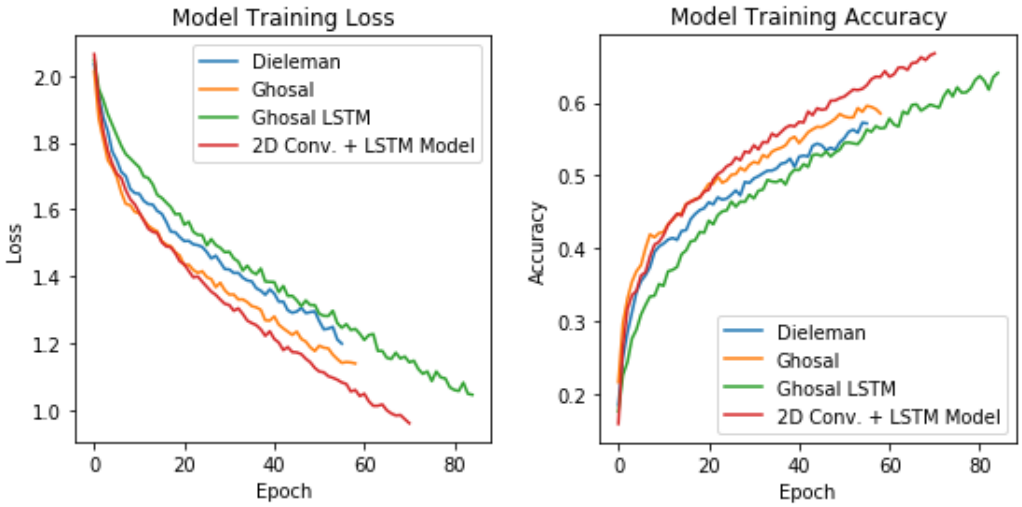

Figure 5: Training loss and classification accuracy of the four models in this paper over the epochs.

very quickly and have very low validation accuracy. We used the validation loss to determine better hyper parameters for our dataset. We hypothesize that the additional hidden units and filters would help with the size of the dataset that Spotify has to offer, but since the FMA dataset is far smaller, the model needs to be simpler.

We reproduced the Ghosal models as closely as possible to the original work [3]. They report validation accuracy as high as 94.2% in [3]. Using their architecture, we were only able to get a validation accuracy as high as 49.50% when reproducing. This discrepancy is likely due to the differences in datasets used to train and test the models. [3] used the GTZAN dataset [9] which contains 10 different genres and 1,000 songs. We used the FMA dataset with 8 genres and 8,000 songs. We speculate that songs in the GTZAN very distinctly belong to singular genres whereas the lines between genre in the songs in the FMA dataset are very blurred.

Replacing the 1D convolutional layers in the Ghosal LSTM model with 2D convolutional layers greatly improved the test accuracy (49.5% to 54.75%). The reason for this increase is that the 1D convolutions allow the model to be too complex. As seen in Figures 5 and 4, the training loss decreases quickly but the validation loss reaches a minimum quickly. When switching to the 2D

convolutional layers, the validation loss still reaches a minimum in about the same time as the 1D convolutions, but it reaches a deeper minimum.

We were also able to determine that it was in fact the switch from 1D convolutions to 2D convolutions that caused the increase in accuracy since we also tested the 2D convolutions in the Ghosal model that did not have an LSTM. This was the second best model that we tried.

We chose to reproduce the Dieleman and Ghosal models due to their high reported accuracies. We reproduced these models and showed that they can be improved by using the 2D convolutions rather than 1D. This shows that our model is the best single network model. [20] reported a test accuracy of 75% using hierarchical ensemble methods. Some of the classifiers that made up the ensembles though utilized 1D convolutional layers. So the classifier in [20] could benefit from the 2D convolutional layer switch that we have investigated in this paper.

In future work, we would like to explore alternatives to the LSTM layer in the classification model. Our data has a lengthy time component to it. The forget gates of an LSTM can help the model draw back on previous time steps to properly classify the music, but the music length might be too long for an LSTM. In [24], they introduce a new model architecture called the Transformer and showed that it can perform extremely well on sequence data such as translation. It would be interesting to see if the Transformer can model the temporal dependencies in the long streams of music better than our architecture.

6 Work Division

For this report, both team members equally contributed to literature reviewing, coding, discussion and final write up.

References

[1] P. Pham A. Ng H. Lee, Y. Largman. Unsupervised feature learning for audio classification using convolutional deep belief networks. 2009.

[2] Y. M. G. Costa D. Bertolini C. N. Silla V. D. Valerio, R. M. Pereira. A resampling approach for imbalanceness on music genre classification using spectrograms. 2018.

[3] M. H. Kolekar D. Ghosal. Music genre recognition using deep neural networks and transfer learning. 2018.

[4] B. Schrauwen A. V. D. Oord, S. Dieleman. Deep content-based music recommendation. 2014.

[5] A. Arya N. M. Karunakaran. A scalable hybrid classifier for music genre classification using machine learning concepts and spark. 2018.

[6] P. Vandergheynst X. Bresson M. Defferrard, K. Benzi. Fma: A dataset for music analysis, 2016.

[7] Colin Raffel Dawen Liang Daniel PW Ellis Matt McVicar Eric Battenberg McFee, Brian and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[8] S. Sigtia and S. Dixon. Improved music feature learning with deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6959–6963, May 2014.

[9] G. Tzanetakis and P. Cook. Musical genre classification of audio signals, 2002.

[10] Grzegorz Gwardys and Daniel Grzywczak. Deep image features in music information retrieval. *International Journal of Electronics and Telecommunications*, 60(4):321 – 326, 2014.

[11] Babu Kaji Baniya, Joonwhoan Lee, and Ze-Nian Li. Audio feature reduction and analysis for automatic music genre classification.

[12] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 141–149, 2017.

[13] A. Foroughmand Arabi and. Enhanced polyphonic music genre classification using high level features. In *2009 IEEE International Conference on Signal and Image Processing Applications*, pages 101–106, Nov 2009.

[14] Y. Panagakis, C. Kotropoulos, and G. R. Arce. Music genre classification via sparse representations of auditory temporal modulations. In *2009 17th European Signal Processing Conference*, pages 1–5, Aug 2009.

[15] Y. Panagakis and C. Kotropoulos. Music genre classification via topology preserving non-negative tensor factorization and sparse representations. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 249–252, March 2010.

[16] B. H. Juang L. R. Rabiner. *Fundamentals of speech recognition*. Prentice Hall, Upper Saddle River, NJ, 1993.

[17] Vinicius Valerio, Rodolfo Pereira, Yandre Costa, Diego Bertoini, and Carlos Silla Jr. A resampling approach for imbalanceness on music genre classification using spectrograms, 2018.

[18] G. J. Qi H. Hu, L. Wang. Learning to adaptively scale recurrent neural networks. 2019.

[19] N. Karunakaran and A. Arya. A scalable hybrid classifier for music genre classification using machine learning concepts and spark. In *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, pages 128–135, March 2018.

[20] S. Hiriyannaiah S. G. Matt K. G. Srinivasa A. Kanavalli M. D. S. Anisetty, G. K. Shetty. Content-based music classification using ensemble of classifiers. 2018.

[21] S. Dieleman. Recommending music on spotify with deep learning, 2014.

[22] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. In *Neural Computation*, volume 9, pages 1735–1780, 1997.

[23] Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[24] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Ashish Vaswani, Noam Shazeer and Illia Polosukhin. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.