Music Genre Classification

Team members: Bingying Liu, Iuliia Oblasova, Zhuangdie Zhou

Abstract

Music genre classification is consisted of two main components, audio features and classifier. This paper implemented a classical long short term memory (LSTM) model on music genre classification on GTZAN dataset and proposed a parallel CNN and RNN model to improve the performance by using various kinds of audio features. After trials on segmentation of the audio files, the proposed Parallel CNN and RNN model reached an accuracy of 72.5%. We found that the architecture of combining CNN and GRU takes into account of both time series feature and melspectrogram image features of the audio file, which results in the accuracy improvement. Additionally, normalization of the input features and audio segmentation largely improved the model performance.

Keywords: Music Genre, Classification, LSTM, CNN

1 Introduction

1.1 Motivation and importance of the problem

Music genre classification has been an active topic of research for the last two decades due to the growing computational power and multiple applications of this research. Spotify and iTunes alone have 73 million songs. An improvement in music genre classification would result in a more accurate recommendation systems for these companies and, consequently, greater revenue rates. Other potential applications include track separation and instrument recognition, music generation, and automatic music transcription [2].

Music genre classification is an ambiguous task because definitions of each genre are not clearly defined, and each song could be a mixture of several genres. However, significant improvements have been made in the last two years and the highest reported accuracy was boosted to 77% by using recurrent neural networks (RNN).[6]

1.2 Approach

In our research, we used GTZAN set – publicly available data which includes 1000 songs of 30 seconds each representing ten musical genres. We experimented with different segment length (30 sec, 5 sec), and a variety of features such as zero-crossing rate etc., detailed description of which will be discussed in the methodology section.

We used librosa package in Python to extract the features and Keras and PyTorch packages for classification. As a baseline models, we used SVM and Logistic Regression which generally are used for classification purposes. Due to the time-series nature of audio recordings, we tried RNN model first. In order to eliminate the drawbacks of RNN such as gradient vanishing and gradient exploding and take advantage of feedback connections, we have implemented LSTM model. Finally, we used parallel CNN RNN model to boost the accuracy of predictions by taking the advantage of both image (mel spectrogram) and time-series information and achieved an accuracy of 72.5%.

2 Related works

While reading the previous publications, we have identified the most commonly used models and the intuitions behind using them. In our research, we attempt to reach the highest reported accuracy for these models by experimenting with different model architectures.

Model	Motivation to use	Highest achieved accuracy in literature
SVM	Baseline model	66.28%
Logistic regression	Baseline model	59.91%
Vanilla LSTM	Prevent the vanishing gradient problem and use feedback connections	65%
Sequential CRNN	Takes advantage of CNNs for local feature extraction and RNNs for temporal summarization of the extracted features. Only used 1000 sample data. Not enough training data.	76%
Parallel CNN and RNN	Involves CNN layers for feature extraction on input data from both visual (mel spectrogram) and audio and combines it with GRU to support sequence prediction.	77%

Figure 1: Music genre classification models used in literature

3 Details of the project

3.1 Data processing

The dataset we used is the most-used public dataset for evaluation in research for music genre recognition (MGR) called Gtzan. It contains 1000 30-second music audio files labeled in ten genres. In order to apply our deep learning methods to the music audio in wav. format, we extracted several different audio features from each of the audio file as our input data.

In general, there are four main categories of audio features: dynamics, rhythmic, spectral, and harmony. They are classified into low-level or high-level according to the frame size. Among those various kinds of features, mel-spectrogram is a high-level feature that represents an acoustic time-frequency representation of a sound. Mel-frequency cepstral coefficients (MFCCs), as a subset feature of mel-Spectrogram, is a low-level spectral feature with short time duration. Figure 2 is a sample of MFCC. While mel-spectrogram and MFCC are the two features we mainly focused on, there are also other features like Spectral Centroid, Chroma, Spectral Contrast that are utilized in our models as supplementary. Figure 1 are 10 samples of mel-spectrogram for each of the 10 genres in the GTZAN data set. We observed that different genres have very different mel-spetrograms, which serves as a motivation for us to explore the visual approach of classification.

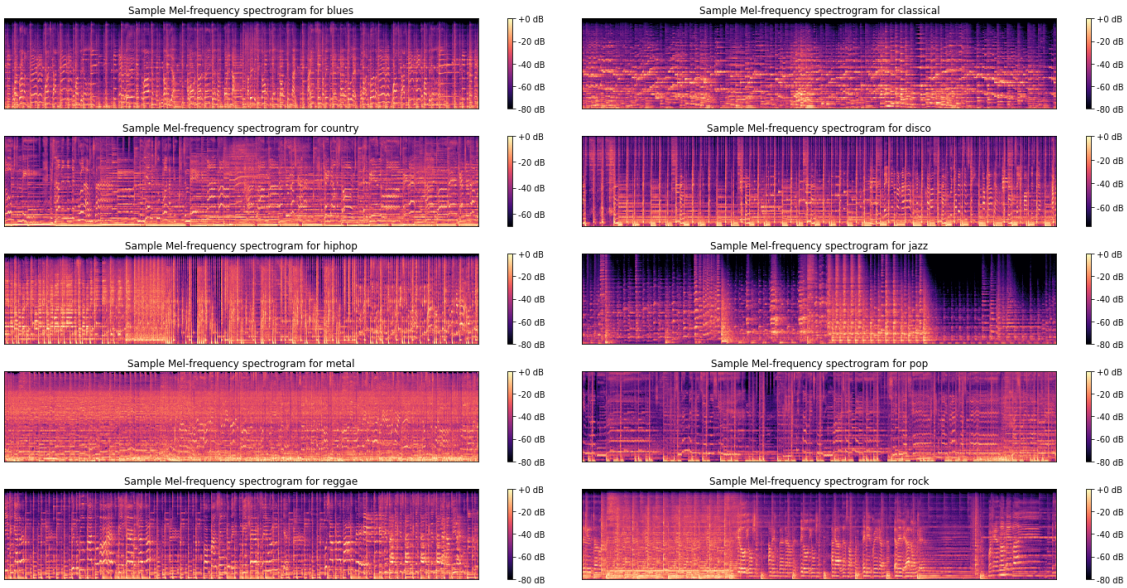

Figure 2: Samples of mel-spectrogram for each genre

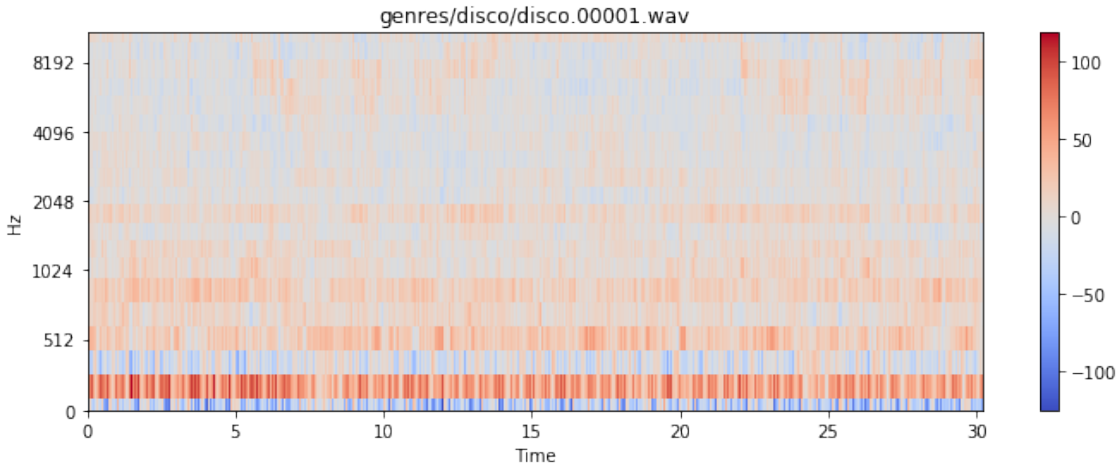

Figure 3: A samples of Mel-frequency cepstral coefficients (MFCCs)

Apart from feature extraction, we segmented the 30-second audio files by different lengths of cut points to extend volume of data. However, by choosing a smaller length, we were able to have more training data. In this way, we extracted more detailed features within smaller pieces of audio instead of features for the whole file, which might not be too general to reflect the local features. Also, we trained our models on different lengths of audios. The intuition was that humans might tell the genre of a song by listening to a shorter length like 5 or 10 seconds, but we were not sure about how long a model needs to recognize genre.

3.2 SVM

We have started with SVM classifier due to its effectiveness in high dimensional spaces: after loading an audio file as a floating-point time series using librosa package, each soundtrack was converted to a list of 661794 elements. [7] In the next step, we extracted the key features characterizing the sound:

- Chromagram: captures harmonic and melodic characteristics of music regardless of timbre and instrumentation.

- Spectral centroid: measure of the brightness of a sound.
- P'th-order spectral bandwidth: the difference between the upper and lower frequencies of a sound.
- Roll-off frequency: the reduction of signal level as the frequency of the signal moves away from the cut-off frequency (in other words, eliminating the frequencies that human cannot hear).
- Zero-crossing rate: rate of sign-changes along a signal (to identify percussive sounds).
- Mel-frequency cepstral coefficient (MFCC): power spectrum of a sound to display the different frequencies present in it. [4]

After extracting these features, we calculated its mean values, so each song is represented by an array of six elements. At the next step, we standardized the values of all features so they all have the same influence on the distance metric used for SVM classification.

With this approach, the average accuracy is 41%. Confusion matrix is presented below.

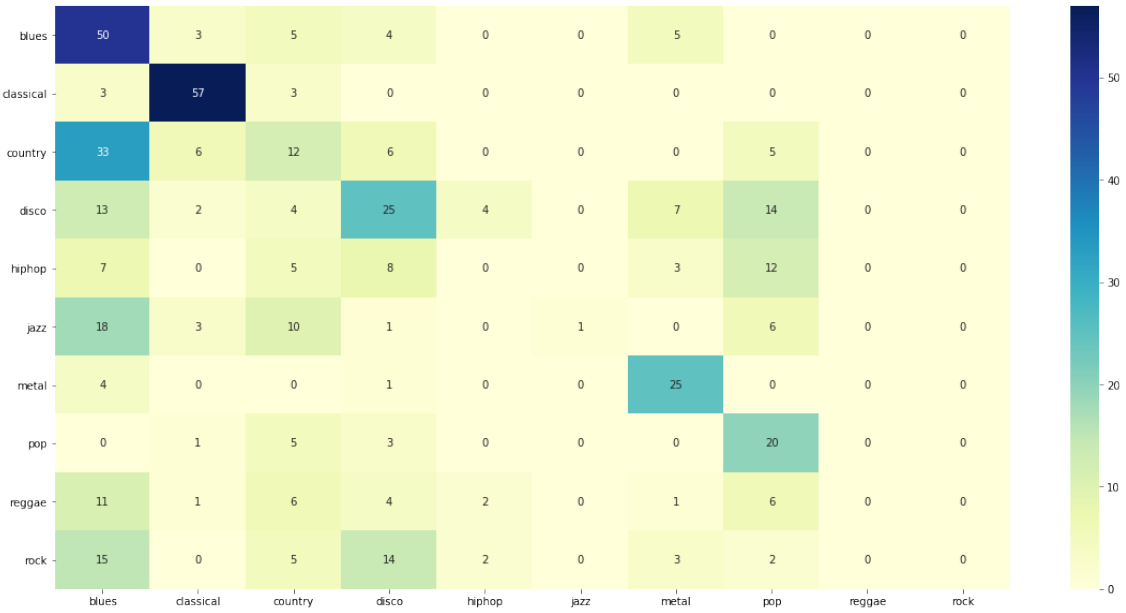

Figure 4: Classification results for SVM model

3.3 Logistic regression

As a baseline model, we also attempted multi-class logistic regression being motivated by its simplicity and well-balanced data set. The results are presented in the table below. Expectedly, the model achieved a low accuracy of 39% on average since the data has a complicated nature and is not linearly separated.

Genre	Precision	Recall	F-1 score	Support
blues	0.24	0.28	0.26	36
classical	0.5	0.91	0.65	22
country	0.4	0.09	0.09	39
disco	0.33	0.13	0.19	38
hiphop	0.5	0.5	0.5	30
jazz	0.56	0.45	0.5	40
metal	0.48	0.9	0.63	31
pop	0.33	0.75	0.46	28
reggae	0.25	0.28	0.26	29
rock	0.14	0.05	0.08	37
accuracy	0.37	0.43	0.36	330
macro avg	0.37	0.39	0.34	330
weighted avg	0.05	0.08	0.08	330

Figure 5: Performance of multiclass logistic regression

Both SVM and Logistic Regression achieved relatively high accuracy for several genres such as metal and classical music and low for rock, disco and country music. In our assumption, it is due to the fact that classical music and metal are well-defined and have strong features specific for these genres. For example, presence in excess of percussive sounds defined by zero-crossing rate well identify metal music, whereas the definition of disco music is fuzzy and do not have any peculiarities.

3.4 Vanilla LSTM

Although we reached 41% accuracy using baseline models, which is way better than random guessing of 10%, we did not include time series features into account in those baselines. And that was how we decided to use LSTM(long short term memory) model for our classification. LSTM is a subset of RNN(recurrent neural networks) used in the field of deep learning. As a model with feedback connections, LSTM is developed to deal with the exploding and vanishing gradient problems that can be encountered by traditional RNNs. Using four different audio features(MFCC, spectral center, chroma and spectral contrast) normalized, we reached an overall accuracy of 65%. The architecture we used for the vanilla LSTM model is shown as below.

3.5 Parallel CNN and RNN

This architecture is heavily inspired by Priyanka Dwivedi's [5] paper on genre classification using Free Music Archived (FMA) dataset. The idea of using a combination of convolution and recurrent neural networks is not unseen in music genre classification and sound event detection. In the methodology

Architecture of Vanilla LSTM	
Input Layer 1	20 MFCC features, 1 Spectral Center features, 12 Chroma features, 7 Spectral Contrast features as input data (normalized)
Hidden Layer 1	128 neurons
Hidden Layer 2	64 neurons
Output Layer	10 outputs corresponding to 10 different music genres

Figure 6: Architecture of Vanilla LSTM

section, we showed that different genres have very different mel-spectrograms,which proves that CNN is a possible approach.

In addition, mel-spectrogram has a time component and GRU is good at capturing short-term and long-term temporal features. As stated in Dwivedi's paper, the intuition for using parallel CNN and RNN approach instead of sequential CNN and RNN approach is that the later one takes in output of CNN as input of RNN. During the process of convolution, original music signals are lost and time sequence is not preserved. Therefore, feeding in mel-spectrograms to both CNN and RNN in parallel, concatenating their outputs and passing it through a dense layer with softmax activation could preserve both image and time series signals.

The convolutional block has 5 2D convolutional layers with ReLu activation and followed by a 2D max pooling layer for each layer except for the last layer. Kernel size used for all layer are (3,1). The first conv2d layer has 16 filters, the second has 32 filters, and remaining layers all have 64 filters. Output of the convolution block is flattened into the shape (None, 384).

The recurrent block starts with a 2D max pooling layer to reduce the size of the mel-spectrogram to speed up RNN processing. The reduced-size images are then fed into a 64- units bidirectional gated recurrent unit. Output of the recurrent block has shape (None, 128).

Outputs from convolutional and recurrent block are then concatenated into a tensor of shape (None, 512). This tensor is then passed into a dense layer with softmax activation. Figure 6 shows the model architecture. We used categorical cross entropy loss as loss function, rmsprop optimizer with a learning rate of 0.005. The model was trained for 50 epochs and learning rate was reduced when valuation accuracy has stopped improving.

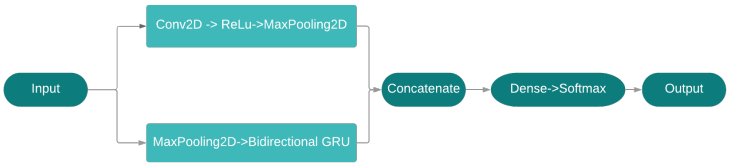

Figure 7: Architecture of Parallel CNN and RNN

We experimented with 1000 29.5-seconds audio as well as the augmented dataset with 6000 4.5-seconds audios as we discussed in the methodology section. Because of the small dataset problem, 29.5-seconds audio does not perform as good as the augmented 4.5-seconds dataset. Dwivedi has suffered from small dataset problem in her experiment, and our approach improves the accuracy. Test accuracy is 54.5% for 29.5-seconds audio and it reaches 72.5% for augmented dataset after fine-tuning the model. Figures 7 and 8 show the model accuracy and loss for the augmented dataset.

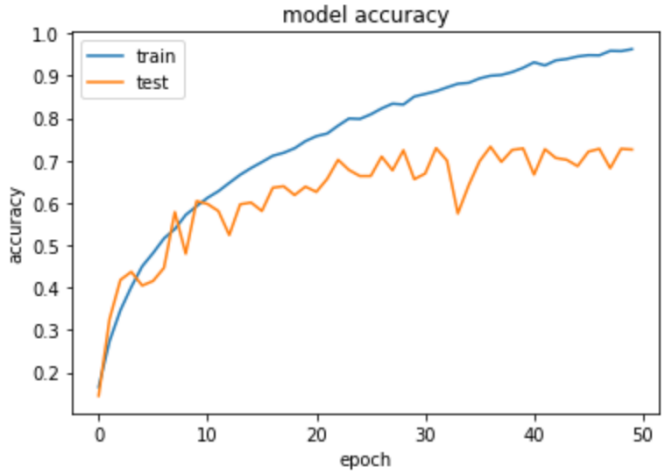

Figure 8: Model accuracy for Parallel CNN and RNN using 6000 4.5s data

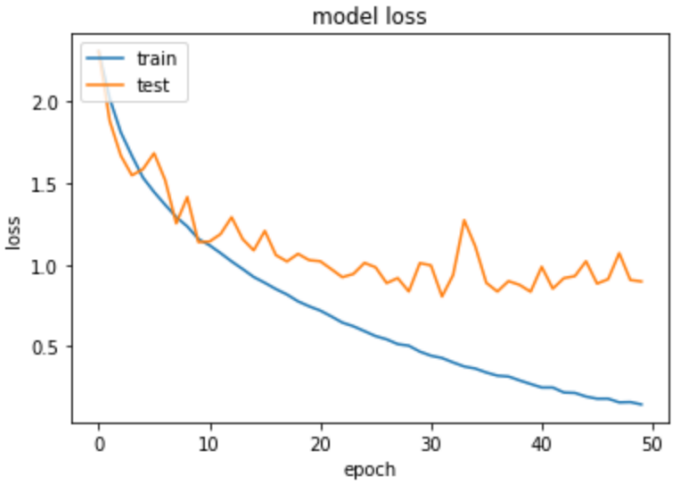

Figure 9: Model loss for Parallel CNN and RNN using 6000 4.5s data

The test curve becomes flat after about 20 epochs and still has relatively high variance, which means it's over-fitting the training set. Figure 8 shows the precision, recall and f1-score for each genre. From the f1-score, we can see that the model classifies metal, classical and jazz pretty well but rock and disco relatively worse.

	precision	recall	f1-score	support
blues	0.75	0.69	0.72	143
classical	0.86	0.92	0.89	125
country	0.72	0.45	0.55	130
disco	0.74	0.55	0.63	115
hiphop	0.66	0.87	0.75	100
jazz	0.86	0.83	0.85	112
metal	0.95	0.92	0.94	118
pop	0.77	0.78	0.78	125
raggae	0.65	0.74	0.69	112
rock	0.41	0.55	0.47	120
accuracy			0.73	1200
macro avg	0.74	0.73	0.73	1200
weighted avg	0.74	0.73	0.73	1200

Figure 10: Precision, recall and f1-score for each genre

From figure 9, we can see that rock is often mislabeled as blues or country and it can sometimes be categorized into other common genres as well. It's because rock has many subgenres that are essentially mixtures of all other genres, such as country rock, blues rock, etc. Therefore, hard-classifying music into one genre is not the optimal way, and one of the future improvements is to get multiple labels for one music and perform soft classification. [9]

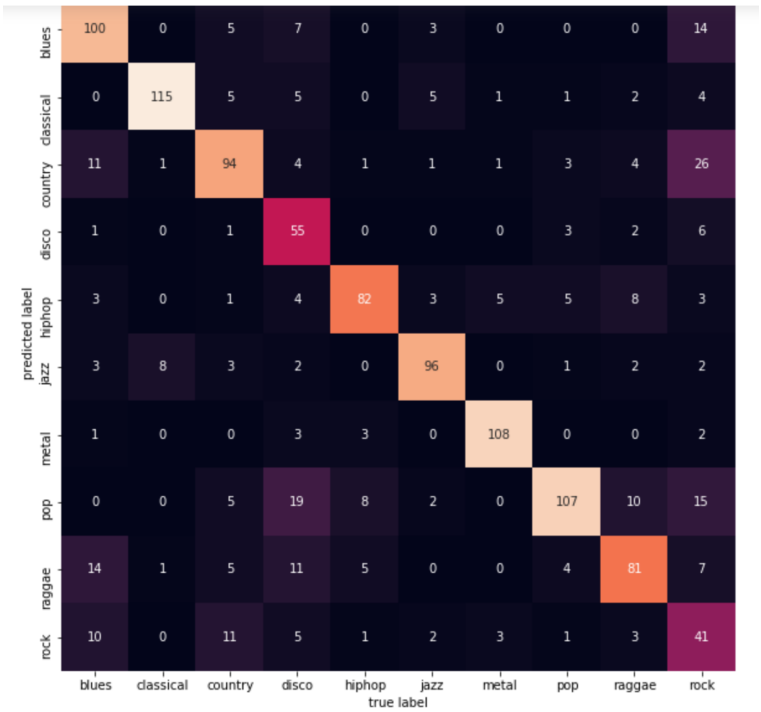

Figure 11: Confusion matrix for Parallel CRNN classification model

3.6 Contribution of each member of the team

Bingying worked on literature reviews, building and tuning parallel CNN and RNN model as well as paper compiling.

Iuliia conducted literature reviews on a considerable number of papers, established two baseline models of logistic regression and SVM and compiled the paper.

Zhuangdie worked on data processing, training and tuning vanilla LSTM model and paper compiling.

4 Experimental results

Figure 11 shows the comparison performance for all models we have implemented. Parallel CNN and RNN model outperforms all other models for most of the genres except reggae and rock for which Vanilla LSTM model performed the best. We do not have an explanation why LSTM performs best for these genres.

Music Genre Classification										
Genre	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Logistic regression	0.26	0.65	0.09	0.19	0.5	0.5	0.63	0.46	0.26	0.08
SVM	0.23	0.65	0.22	0.17	0.56	0.5	0.57	0.36	0.22	0.3
Vanilla LSTM	0.36	0.82	0.47	0.58	0.55	0.75	0.45	0.72	0.94	0.86
Parallel CNN and RNN	0.72	0.89	0.55	0.63	0.75	0.85	0.92	0.78	0.74	0.55

Figure 12: Figure 3. Model comparison for each genre

5 Concluding remarks and future works

From the performance of our model, we can see that deep learning methods are useful and handy for music genre classification. However, the misclassifications also indicate that it might not be most appropriate to assign only one label to one song. Soft classification would be more reasonable in some cases. And that is what could be explored next.

Nevertheless, if we only focus on further improving the classification accuracy, there are several aspects that can be considered.

First, the dataset GTZAN we used has its natural faults including repetitions, mislabelings, and distortions. There is research showing that there are misclassifications in the GTZAN dataset, which will definitely have a negative impact on genre classification. If we could fix some of the misclassified genres of the dataset, better performance could be achieved.[8][3]

Second, the features extracted from a single audio file has different scales, which results in biased weights when we stack features together. For instance, if all the scale of one feature is much larger than another, the effect on the classification of the feature with smaller scale will vanish. Therefore, recurrent batch normalization might be a feasible way to deal with this problem.

Third, we currently initialize our weights by random, but there are several approaches to initialize weights scientifically. This will not only increase our training speed but will also avoid weight vanishing if local minimum is reached.

Finally, instead of using a CNN from scratch we could try using pre-trained models like VGG or Inception when conducting parallel CNN and RNN. [10][1]

References

[1] Long short-term memory. Wikipedia.

[2] Music information retrieval. Wikipedia.

[3] Bozena Kostek Aldona Rosner. Automatic music genre classification based on musical instrument track separation. SpringerLink, 2017.

[4] Librosa development team. Librosa package documentation: feature extraction.

[5] Priyanka Dwivedi. Deep learning for music recognition. 2018.

[6] Nadav Hollander Jeremy Irvin, Elliott Chartock. Recurrent neural networks with attention for genre classification. SemanticScholar, 2016.

[7] Armando Stellato Roberto Basili, Alfredo Serafini. Classification of musical genre: a machine learning approach. ResearchGate, 2013.

[8] Rongbin Li Scott Zhang, Huaping Gu. Music genre classification: near-real time vs sequential approach. 2019.

[9] Francesco Barbieri Sergio Oramas. Multimodal deep learning for music genre classification. TISMIR, 2018.

[10] Gonzalo R. Arce Yannis Panagakis, Constantine Kotropoulos. Music genre classification via sparse representations of auditory temporal modulations. IEEE, 2009.