Music Genre Classification

Ruochen Kong
ruochen.kong@emory.edu
Emory University
Atlanta, Georgia, USA

ABSTRACT

Searching music by genre is a function implemented in almost all music apps, so developing an accurate algorithm to classify the music is crucial. The existing research papers have successfully classified music into no more than 10 genres, but in the real world, much more detailed genres exist. This rough classification may further cause an unsatisfaction with the user experience. In this project, I investigated the signal of the music in 19 genres, extracted distinguishable features, and applied them to classification models. The result shows that, in general, the simple 2-layer CNN performs better than Random Forest, the normalized features perform better than original features, and a selection of features based on correlations improves the performance.

KEYWORDS

Signal Processing, Data Mining, Machine Learning, Music Analysis

1 INTRODUCTION

Major music applications, such as Spotify, Apple Music, or Gnoosic, have all allowed users to search by genre. The accuracy of the search result highly affects the satisfaction of users. The previous research has successfully developed models to classify music into 10 different genres with accuracy higher than 80% evaluated on the GTZAN Dataset [6, 7, 11, 13]. These models, however, may not be as successful when facing more genres of datasets in the real world. According to Tamatjita et al. (2016), the performance of classification models drops significantly by increasing the number of genres from 6 to 12 [14]. This drawback is understandable because the more the genres the more demanding the model is to detect differences between features, but pursuing a more accurate automatic classification method is still necessary.

This project aims to establish a method to classify music into 19 genres with the dataset found on Kaggle[1]. The dataset contains 19.9k music labeled as *Electronic, Rock, Punk, Experimental, Hip-Hop, Folk, Chiptune and Glitch, Instrumental, Pop, International, Ambient Electronic, Classical, Old-Time and Historic, Jazz, Country, Soul-RnB, Spoken, Blues, or Easy Listening*. To develop the model, I started with extracting features from raw signal data. Common audio features are considered, including beats, tuning, zero-crossing rate, autocorrelation, root mean square energy (RMS), centroid, flatness, Mel-frequency cepstral coefficient (MFCC), and short-time Fourier transformation (STFT) [10, 12, 16]. The extracted features are normalized and filtered by correlations [18], and then applied with random forest (RF) and a 2-layer CNN. The overall accuracy of each model is higher than 45% with the highest accuracy 47.6% achieved by CNN with features both normalized and filtered. Hence, by comparison of the use of different models and different feature formats, a general conclusion is that CNN outperforms RF and normalized features outperform the original features.

The rest of the report is organized as: Section 2 presents the related work on music classification based on audio signals. Section 3 provides an overview of the dataset. Features used in this project are extracted and selected in Section 4. Section 5 provides the experiment process and the results. The discussion of limitations and possible improvements is in section 6. A conclusion of the project is finally provided in section 7.

[1] https://www.kaggle.com/competitions/kaggle-pog-series-s01e02

2 RELATED WORK

In 2001, Tzanetakis et al. presented groundbreaking research in music classification [16]. The authors summarized the music features into three general types, (i) timbral texture features, (ii) rhythmic content features, and (iii) pitch content features. Timbral texture features include spectral centroid, spectral roll-off, spectral flux, zero crossings, MFCC, analysis windows, and low-energy features; rhythmic content features are mainly beat analysis by full-wave rectification, low-pass filtering, downsampling, mean removal, enhanced autocorrelation, peak detection, and histogram calculation, and beat histogram features; pitch content features are based on multiple pitch detections with the technic provided by Tolonen and Karjalainen in 2000 [15]. The authors also developed a dataset, GTZAN, which is labeled hierarchically that allows classification tasks with both 10 genres and 20 genres. The GTZAN dataset has then been widely used in evaluations of the following research.

Several models have been developed and evaluated on the GTZAN dataset. Lee et al. (2009) and Fu et al. (2010) achieved classification accuracy slightly higher than 90% both with MFCC, amplitude spectrum envelop (ASE) and octave based spectral contrast (OSC) [5, 7]. Li et al. (2010) reported accuracy of 84% by CNN with majority votes [8]. Feng (2014) reported accuracy of 61% with a 5-layer Restricted Boltzam Machine (RBM) [4].

Finally, Tamatjita et al. (2016) presented a comparison of different numbers of genres to be classified [14]. The authors used Nearest Centroid Classifier as their model with the same audio features to classify music into 3, 6, 9, and 12 genres. They used 120 songs as test data and reported an accuracy of 96.7% on 3 genres, 70% on 6 genres, 53.3% on 9 genres, and 33.3% on 12 genres.

3 DATASET

The dataset contains 19922 ogg files with 3071 in Electronic, 3095 in Rock, 2582 in Punk, 1800 in Experimental, 1757 in Hip-Hop, 1241 in Folk, 1181 in Chiptune and Glitch, 1044 in Instrumental, 945 in Pop, 814 in International, 796 in Ambient Electronic, 495 in Classical, 408 in Old-Time and Historic, 306 in Jazz, 142 in Country, 94 in Soul-RnB, 94 in Spoken, 58 in Blues, and 13 in Easy Listening. The corresponding genre ids are in the same order starting from 0. The genre was annotated by humans which may include errors.

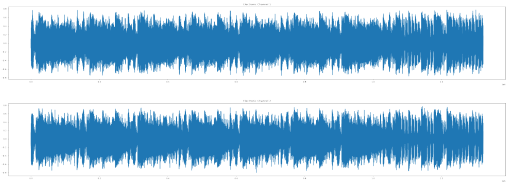

Figure 1: Electronic

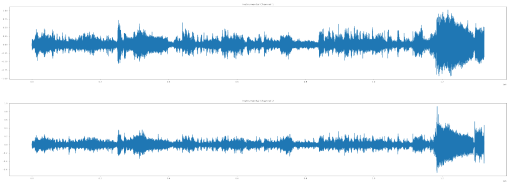

Figure 2: Instrumental

Each music segment is around 30 seconds with slight differences and contains 2 channels. Figure 1 and 2 represents an example of electronic music and instrumental music.

4 FEATURES

Common audio features are used in this project, including beats, tuning, zero-crossing rate, autocorrelation, RMS, centroid, flatness, MFCC, and STFT.

4.1 Feature Extraction

The entire feature extraction process was completed with the LibROSA library [9]. The audios are loaded into signals with a sampling rate equal to 22050 and all the signals are resized into lengths equal to 660000 by either padding with 0 or clipping. The correlation between two channels is first calculated, and the following features are extracted only from the first channel. A series of audio signal s can be decomposed into the sum of harmonic signals, s_h, with persuasive signals, s_p, namely $s = s_h + s_p$.

Beats number and zero-crossing rate is then extracted from s, s_h and s_p. Estmimated tuning is calculated for s. RMS, centroid, and flatness are calculated from all three signals. The mean and standard deviation values of these three features generated from all three signals are collected with additionally the max and min values generated from s only. 20 MFCCs are calculated from s, and the max, min, mean and standard deviation values are collected. STFT is calculated with FFT window size equals 8 and the number of chroma equals 12. Only the first 20 columns of STFT are collected due to the consideration of computational costs. Finally, autocorrelations of s are calculated with 4 different starting points.

Table 1 shows the number of features extracted from each audio.

4.2 Feature Selection

Considering the Random Forest will be used to classify the data points, the number of features would affect the performance. Thus, the following steps aimed to investigate the distinguishability of features and the correlations between each feature.

	num of Features
Beats, Zero Crossing Rate	$3 \times 2 = 6$
Tuning	1
RMS, Centroid, Flatness	$2 \times (3 \times 3) + 2 \times 3 = 24$
MFCC	$20 \times 4 = 80$
STFT	$12 \times 20 = 240$
Autocorrelation	4
Overall	**356**

Table 1: Summary of extracted features

The value of a feature should vary between genres in order to allow the model to learn the difference otherwise it would be helpful. Two box plots are generated for each of the main features to investigate the effectivity with one including outliers and one that does not. The main features include the beat, tuning, zero-crossing rate, correlation, as well as the mean value of RMS, centroid, and flatness. As shown in Figure 3, beats do not show an obvious difference among most genres, while the remainings do, for example, the zero-crossing rate shown in Figure 4. According to the consideration that beats are the fundamental feature of music, this feature has still remained. Based on the box plots, the extracted features are generally effective in the classification task.

Besides the consideration of distinguishability of features, the correlation between features is another concern in selecting features. The goodness of a feature in the classification task can be summarized as the feature should be highly correlated with the

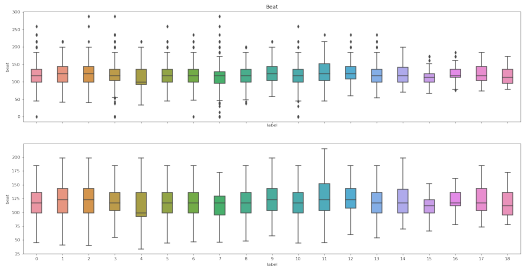

Figure 3: Beats: Feature without obvious different

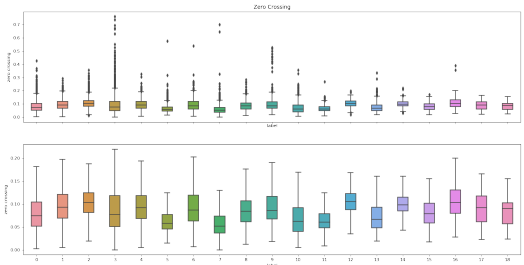

Figure 4: Zero Crossing Rate: Feature with obvious different

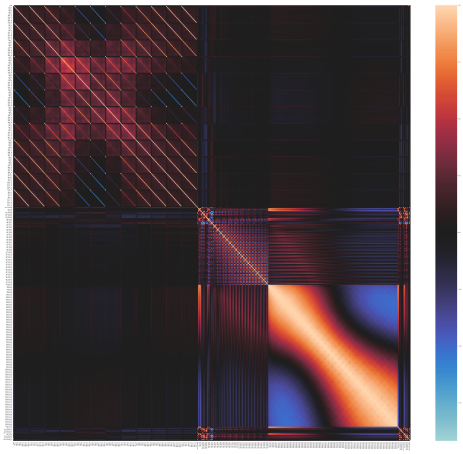

Figure 5: Heatmap representation of correlations

genre_id	**Norm**	**Origin**	**Filter**
0	**0.146819**	0.108962	0.137913
1	0.119828	0.120253	**0.137296**
2	0.141254	0.120257	**0.186090**
3	0.098007	**0.128776**	0.080893
4	**0.192417**	0.099583	0.161937
5	0.108278	0.095206	**0.153144**
6	0.065004	0.057158	**0.098481**
7	0.099402	0.075729	**0.108641**
8	**0.056432**	0.045413	0.049636
9	0.085329	0.056055	**0.086779**
10	**0.057931**	0.040620	0.054520
11	**0.143411**	0.055924	0.118032
12	0.292079	0.047026	**0.473267**
13	0.030270	0.021597	**0.0608**
14	0.016350	0.009717	**0.019933**
15	0.011255	0.009434	**0.014237**
16	0.016536	0.010711	**0.023174**
17	0.005904	0.004182	**0.006872**
18	0.003693	0.002240	**0.004418**S

Table 3: Jaccard scores of k-means

labels while detached from other features [18]. Yu and Liu (2003) provided two effective methods to solve the problem of correlation, but due to the limits of computational power, in this project, the correlated features are filtered simply by randomly selecting one feature from the pair with a correlation higher than 0.8. Figure 5 is the heatmap of the correlation between features. The bright red and bright blue area represents the questionable features with high correlations. In genreral, autocorrelations are highly correlated which forms the most questionable pairs. The columns of STFT and the standard deviation of MFCCs are also hightly correlated.

4.3 Generate Preprocessed Dataset

Denote the dataset with 356 features generated from each audio as **Origin**. Beacuse the original value of features varies from 10^{-2} to 10^3, a normalization of the features would improve the performance by shifting the attention away from the features with larger values. Assume $\mathbf{x}$ is a feature, then the normalization is achieved by:

$$\mathbf{z} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$$

After normalizing with this equation, values of all features are in the range of 0 to 1. Denote the normalized dataset as **Norm**. Finally, the highly correlated features are filtered from the normalized dataset, and the remaining dataset is stored as **Filter**. The dimensions of these three datasets are listed in Table 2.

	Dim	Range
origin	(19909, 356)	(0.01, 3200)
norm	(19909, 356)	[0,1]
filter	(19909, 193)	[0,1]

Table 2: Dimension and Range of each dataset

In order to further analysis whether the normalization and filteration affect the performance of classification, the three datasets are applied to a unsupervised clustering method, k-means. The results from k-means are compared with the ground truth label with the Jaccard score, where

$$Jaccard = \frac{|X \cap Y|}{|X \cup Y|}.$$

The results are shown in Table 3. From the results, the normalized dataset outperforms the original one, and the filtered dataset outperforms the non-filtered one. Thus, I would expect that the **Filter** outperforms others in the supervised classification task.

5 EXPERIMENTS

The three datasets are splited into training and testing pair with propotion 80%/20% in the same way, namely all the training datasets are the features from the same group of raw data, as well as the testing datasets. Each training dataset has 15,927 data points and each testing dataset has 3,982 data points. Specificlly, the testing dataset has 644 in Electronic, 627 in Rock, 554 in Punk, 341 in Experimental, 311 in Hip-Hop, 250 in Folk, 226 in Chiptune and Glitch, 210 in Instrumental, 185 in Pop, 162 in International, 163 in Ambient Electronic, 86 in Classical, 80 in Old-Time and Historic, 57 in Jazz, 32 in Country, 17 in Soul-RnB, 18 in Spoken, 17 in Blues, and 2 in Easy Listening.

5.1 Models

Two models are used in this project, Random Forest and CNN. In order to improve the performance of RF, 10 different RF are grown on the same training data to predict the testing data. The final result is formed from a majority vote. The CNN is structured with 2 Conv1D layers, a batch normalization layer, a dropout layer with

genre_id	RF_origin	RF_origin	RF_origin	RF_origin	RF_norm	RF_norm	RF_norm	RF_norm	RF_filter	RF_filter	RF_filter	RF_filter	CNN_origi	CNN_origi	CNN_origi	CNN_origi	CNN_norm	CNN_norm	CNN_norm	CNN_norm	CNN_filter	CNN_filter	CNN_filter	CNN_filter
	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall
0	0.7674535	0.5021505	0.3840461	0.7251553	0.7694626	0.5032468	0.3862126	0.7220497	0.7767454	0.513942	0.3966245	0.7298137	0.8013561	0.5173887	0.4261307	0.6583851	0.7674535	0.5172054	0.389325	0.7701863	**0.8242089**	0.5388669	0.4679634	0.6350932
1	0.78001	0.4455696	0.3693599	0.5614035	0.7752386	0.4360428	0.3604167	0.5518341	0.7737318	0.4434836	0.3618952	0.5725678	0.7388247	0.3048128	0.2623705	0.3636364	**0.8277248**	0.3042596	0.4178273	0.2392344	0.7847815	0.4166099	0.3634204	0.4880383
2	0.8799598	0.5082305	0.5909091	0.4458484	0.8776996	0.5035678	0.5784543	0.4458484	**0.884229**	0.5252317	0.6115108	0.4602888	0.8184329	0.499654	0.4051627	0.6516245	0.8458061	0.5633001	0.4647887	0.7148014	0.8553491	0.5278689	0.4834835	0.5812274
3	0.8960321	0.3669725	0.3833866	0.3519062	0.8952788	0.3672231	0.3805031	0.3548387	0.8997991	0.4035874	0.4115854	0.3958944	0.8375188	0.3281412	0.2540193	0.4633431	0.9103466	0.4157119	0.4703704	0.372434	**0.9178805**	0.4410256	0.5286885	0.3782991
4	0.9372175	0.5791246	0.6077739	0.5530547	0.9369663	0.5781513	0.6056338	0.5530547	0.9427423	0.6068966	0.6542751	0.5659164	0.9367152	0.4220183	0.736	0.2958199	0.9399799	0.6010017	0.625	0.5787781	**0.946007**	0.6140036	0.695122	0.5498392
5	0.9394776	0.5561694	0.5153584	0.604	0.9417378	0.5671642	0.5314685	0.608	0.9412356	0.5714286	0.527027	0.624	0.9447514	0.2903226	0.75	0.18	**0.9480161**	0.5490196	0.6028708	0.504	0.9455048	0.5786408	0.5622642	0.596
6	**0.9588147**	0.4605263	0.8974359	0.3097345	0.9578101	0.4437086	0.8815789	0.2964602	0.9600703	0.488746	0.8941176	0.3362832	0.9482672	0.2076923	0.7941176	0.119469	0.9568056	0.4451613	0.8214286	0.3053097	0.9547966	0.5187166	0.6554054	0.4292035
7	0.9367152	0.34375	0.3793103	0.3142857	0.9352084	0.3246073	0.3604651	0.2952381	0.9342039	0.2880435	0.335443	0.252381	0.9173782	0.3014862	0.2720307	0.3380952	0.9183827	0.3380855	0.2953737	0.3952381	**0.938222**	0.3387097	0.3888889	0.3
8	**0.9540432**	0.0213904	1	0.0108108	**0.9540432**	0.0213904	1	0.0108108	0.9537921	0.0107527	1	0.0054054	0.9532898	0	0	0	0.9527875	0	0	0	0.9510296	0.109589	0.3529412	0.6648649
9	0.9630839	0.3098592	0.6470588	0.2037037	0.9628327	0.3148148	0.6296296	0.2098765	**0.9638373**	0.3076923	0.6956522	0.1975309	0.9593169	0.0121951	0.5	0.0061728	0.9495229	0.3619048	0.372549	0.3518519	0.9457559	0.3757225	0.3532609	0.4012346
10	**0.9588147**	0.0120482	0.3333333	0.006135	**0.9588147**	0.0120482	0.3333333	0.006135	0.9585635	0.011976	0.25	0.006135	0.9585635	0	0	0	0.9580613	0.1522843	0.4411765	0.0920245	0.9540432	0.1793722	0.3333333	0.1226994
11	0.9821698	0.6282723	0.5714286	0.6976744	0.9821698	0.6243386	0.5728155	0.6860465	0.9816675	0.617801	0.5619048	0.6860465	0.9831743	0.4806202	0.7209302	0.3604651	**0.9846811**	0.6347305	0.654321	0.6162791	0.9751381	0.5751073	0.4557823	0.7790698
12	**0.9972376**	0.9325153	0.9156627	0.95	**0.9972376**	0.9325153	0.9156627	0.95	**0.9972376**	0.9325153	0.9156627	0.95	0.9957308	0.8917197	0.9090909	0.875	**0.9972376**	0.9271523	0.9859155	0.875	0.9967353	0.9192547	0.9135802	0.925
13	0.9866901	0.1311475	1	0.0701754	0.9866901	0.1311475	1	0.0701754	**0.9869412**	0.1612903	1	0.0877193	0.9856856	0	0	0	0.9824209	0.0789474	0.1578947	0.0526316	0.9849322	0.1428571	0.3846154	0.0877193
14	**0.9919638**	0	0	0	**0.9919638**	0	0	0	**0.9919638**	0	0	0	**0.9919638**	0	0	0	**0.9919638**	0	0	0	0.9914616	0.0555556	0.25	0.03125
15	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0
16	**0.9969864**	0.5	1	0.3333333	**0.9969864**	0.5	1	0.3333333	0.9964842	0.4615385	0.75	0.3333333	0.9954797	0	0	0	0.9959819	0.2727273	0.75	0.1666667	0.9954797	0.4	0.5	0.3333333
17	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0	0.9957308	0	0	0
18	0.9994977	0	0	0	0.9994977	0	0	0	0.9994977	0	0	0	0.9994977	0	0	0	0.9994977	0	0	0	0.9994977	0	0	0

Figure 6: "One over rest" evaluations for 19 genres on 2 models and 3 feature sets

a 0.5 dropout rate, a flatten, and then a dense layer to shape (,100), and finally the output layer with output shape (,19).

5.2 Evaluation Metrices

The evaluation is based on accuracy, F1_score, precision, recall and AUC score. Evaluations of both overall (Table 4) and labels indivisually (Figure 6) are provided.

	ACC	F1	Precision	Recall	AUC
RF_origin	0.4588	0.3315	**0.5050**	0.3230	
RF_norm	0.4555	0.3295	0.5019	0.3207	
RF_filter	0.4671	0.3339	0.4929	0.3265	
CNN_Origin	0.3787	0.2240	0.3174	0.2269	0.8311
CNN_Norm	0.4591	0.3243	0.3920	0.3176	0.8616
CNN_filter	**0.4761**	**0.3543**	0.4047	**0.3528**	**0.8707**

Table 4: General Evaluations

All the evaluation values are calculated as the average of the performance over 5 times execution. The confusion matrix of the best accuracy example of each model and feature set combination is also generated, for example, the result of CNN_filter is shown in Figure 7.

6 DISCUSSION

As shown in Figure 6, CNNs generally perform better than RFs. Within CNNs, the filtered feature set performs the best, then follows the normalized feature set, and the original one performs the worst. Among RFs, though using filtered features shows slight improvement, no apparent differences are presented in the results from original and normalized feature sets. This invariance may be due to the specific property of RF that it makes decisions based on tree-like formate instead of based on weights in linear models. Moreover, the last 5 lines, representing genres Country, Soul-RnB, Spoken, Blues, and Easy Listening, are mostly zeros representing the lack of capability in classifying these genres. This limitation may be because of the lack of training data in these genres, namely the number of all these genres is only about 2% of the entire training dataset. This problem could be solved by either collecting more music from these genres or merging them into other more common genres.

The best accuracy achieved in this project is around 47.5%. Although this accuracy is better than several models in classifying music into more than 10 genres [14], and its AUC score is similar to that of the CRNN model tasked to classify into 20 genres [1], it is much lower than the accuracy higher than 80% achieved by previous research on GTZAN database [5, 7, 8]. This relatively low performance would be explained by several reasons, including (i) the genres are human-annotated which contain errors, (ii) the dataset is highly imbalanced on several genres, (iii) the possible features are not fully contained, for example, STFT and MFCCs, because of the consideration of the computational capability of the device and (iv) the structure of the CNN is too simple to be robust.

Besides the problem on the dataset itself, others could be improved in future work. The STFT and MFCC features could be pre-trained with CNN to extract features to reduce the dimension [2, 6], instead of achieving the goal by simply dropping the columns in this project. Other audio features could also be considered, for example, the rhythmic content features promoted by Tzanetakis et al. (2002) [16]. In [17], Valero and Alias announced that the Gammatone Cepstral Coefficients (GTCC) is more powerful than MFCC

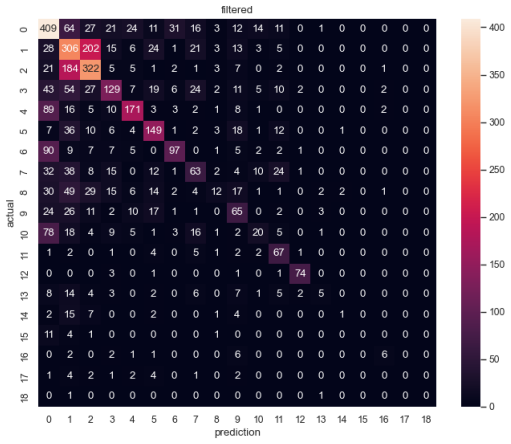

Figure 7: Confution Matrix of CNN_filter

when facing classification tasks, so GTCC may be investigated in the future research. Moreover, according to Dielman and Schrauwen (2014), the pooling layer is inappropriate in this music classification context [3]. Thus, the CNN structure could be improved by removing the pooling layer, adding more convolutional layers or dense layers, or changing the shape of outputs in the hidden layers.

7 CONCLUSION

This project aimed to develop models on classify music into more than 10 genres. The dataset used in this project is found on Kaggle, which contains 19.9 labeled music segments in the 30s. Audio signal features are extracted from the raw data, and, by different pre-processing steps, transformed into three different feature sets. Two models, Random Forest and 2-Layer CNN were then built to classify the music based on these feature sets and achieved the best accuracy higher than 47%. After investigating the evaluations, the limitations of this project are summarized. The limitations include (i) the questionable label quality and the imbalance that occurs in the raw dataset, (ii) the dropoff during the feature extraction, and (iii) the structure of the classifiers. Future research will aim to solve this problem and achieve a better-performed model on music genre classification.

REFERENCES

[1] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2392–2396.

[2] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. 2011. Audio-based music classification with a pretrained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*. University of Miami, 669–674.

[3] Sander Dieleman and Benjamin Schrauwen. 2014. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6964–6968.

[4] Tao Feng. 2014. Deep learning for music genre classification. *private document* (2014).

[5] Zhouyu Fu, Guojun Lu, Kai-Ming Ting, and Dengsheng Zhang. 2010. On feature combination for music classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 453–462.

[6] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. 2010. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia* 13, 2 (2010), 303–319.

[7] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Hwai-San Lin. 2009. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Transactions on Multimedia* 11, 4 (2009), 670–682.

[8] TL Li, Antoni B Chan, and AH Chun. 2010. Automatic musical pattern feature extraction using convolutional neural network. *Genre* 10, 2010 (2010), 1x1.

[9] Brian McFee, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, Colin Raffel, Frank Zalkow, Ayoub Malek, Dana, Kyungyun Lee, Oriol Nieto, Dan Ellis, Jack Mason, Eric Battenberg, Scott Seyfarth, Ryuichi Yamamoto, viktorandreevichmorozov, Keunwoo Choi, Josh Moore, Rachel Bittner, Shunsuke Hidaka, Ziyao Wei, nullmightybofo, Adam Weiss, Darío Hereñú, Fabian-Robert Stöter, Pius Friesch, Matt Vollrath, Taewoon Kim, and Thassilo. 2022. *librosa/librosa: 0.9.1*. https://doi.org/10.5281/zenodo.6097378

[10] Meinard Muller, Daniel PW Ellis, Anssi Klapuri, and Gaël Richard. 2011. Signal processing for music analysis. *IEEE Journal of selected topics in signal processing* 5, 6 (2011), 1088–1110.

[11] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. 2009. Music Genre Classification Using Locality Preserving Non-Negative Tensor Factorization and Sparse Representations.. In *ISMIR*. Citeseer, 249–254.

[12] Geoffroy Peeters. 2004. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Ist Project Report* 54, 0 (2004), 1–25.

[13] Siddharth Sigtia and Simon Dixon. 2014. Improved music feature learning with deep neural networks. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 6959–6963.

[14] Elizabeth Nurmiyati Tamatjita and Aditya Wikan Mahastama. 2016. Comparison of music genre classification using Nearest Centroid Classifier and k-Nearest Neighbours. In *2016 International Conference on Information Management and Technology (ICIMTech)*. 118–123. https://doi.org/10.1109/ICIMTech.2016.7930314

[15] Tero Tolonen and Matti Karjalainen. 2000. A computationally efficient multipitch analysis model. *IEEE transactions on speech and audio processing* 8, 6 (2000), 708–716.

[16] G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10, 5 (2002), 293–302. https://doi.org/10.1109/TSA.2002.800560

[17] Xavier Valero and Francesc Alias. 2012. Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia* 14, 6 (2012), 1684–1689.

[18] Lei Yu and Huan Liu. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*. 856–863.