Parallel Recurrent and Convolutional Neural Networks for Music Genre Classification

Mona Fadaviardakani
University of British Columbia
Vancouver, Canada
mfadavi@cs.ubc.ca

Kevin Chow
University of British Columbia
Vancouver, Canada
kchowk@cs.ubc.ca

Jeffrey Goh
University of British Columbia
Vancouver, Canada
gohzenhao@gmail.com

ABSTRACT

Due to ambiguity between the boundaries of genre labels, music genre classification has always been a challenging but important task in the field of music information retrieval. In this project, we replicate and extend the promising results of the parallel recurrent and convolutional neural networks (PRCNN) approach with the recently published Free Music Archive (FMA) dataset. Key extensions to the approach include the use of log-mel spectrograms and attention-based mechanisms. We achieved a test accuracy score of 45% using our PRCNN implementation, classifying music into eight different genre labels.

Author Keywords

Music genre classification, deep learning, spectrograms, CNN, RNN, attention

INTRODUCTION

With the recent surge in deep learning research, many new applications for neural networks have emerged. One such application is music genre classification (MGC), which has important value in terms of both pure research and commercial applications (e.g., Spotify, Shazam, etc.).

The genre of a piece of music is a useful descriptor that is used in a multitude of ways: as a search query, in the organization of large collections of music, and for recommending new songs to avid music listeners. Thus, it is critical that music-centric applications can both quickly and accurately classify genre.

For music streaming services like Spotify, a large amount of time and research has been put into their genre classification algorithms [8]. According to the website *Every Noise at Once*, there are about 3,796 unique genres as of this moment [1]. With the growing number of genres and the challenges associated with the task, MGC has often become reserved for those who can develop algorithms based on musical features, requiring both significant musical and computational expertise. In some cases, genres may even need to be classified manually, with human intervention.

ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

But, with an increase in freely available and labeled music datasets online, like the Free Music Archive [2], deep learning presents promising new approaches for MGC. One such approach is a parallel recurrent and convolutional neural network (PRCNN), first proposed by Feng et. al. [6] in 2017, which was trained and tested on the GTZAN dataset [12].

In this project, we replicate and extend the promising results of Feng's PRCNN approach. We (1) re-implement their described architecture, while training and testing on the newer, more up-to-date FMA dataset, (2) propose several key tweaks in hopes of improving accuracy, and (3) begin to extend their work by looking at attention-based mechanisms. We also perform several ablation studies to investigate the utility of each component in the PRCNN architecture, and finally, discuss our findings, lessons learned, and potential future work.

By applying a similar approach to MGC as Feng et al.'s [6], but on a different dataset, we investigate and report on the replicability and generalizability of their results, which is an important, but often underappreciated, contribution to machine learning research.

RELATED WORK

Feature-based Approaches

Machine learning techniques have been widely used for music genre classification. Following are some examples. Tzanetakis et al. [13] used hand crafted features including timbral texture, rhythmic content and pitch content along with k-nearest neighbors algorithms for genre classification. Costa et al. [4] considered both acoustic features and audio spectrograms for music classification. They used their experiments on three datasets including ISMIR 2004 Database, LMD database, and a collection of field recordings of ethnic African music. Mutiria et al. [11] classified 31 different combination of feature sets using three different kernels in SVM classifier for non linear data. They used combination of different feature sets of Musical Surface, MFCC, Rhythm, tonality and LPC.

Deep Learning-based Approaches

With the huge success of deep neural networks, a number of studies apply these techniques to audio data as well [9]. Although, it is useful using audio representation in time domain, it is not very easy to work with huge number of audio samples. Since spectrograms varies from genre to genre, some approaches benefit from spectrograms besides the amplitudes and using convolution neural networks to distinguish the genre

patterns. Choi et al. [3] used 1 dimensional CNN for local feature extraction from spectograms and also a recurrent neural networks for temporal summarising of the extracted features. They conducted their experiments on GTZAN dataset.

Since different temporal steps of a song have different importance to decide which genre this song belongs to, some approaches use attention mechanism to find the focus points. Yanug et al. [14] incorporated attention mechanism into representation learning of a music by only focusing on parts of spectrogram's that has the most energy. Music spectrograms are firstly encoded into vector representations by using bi-directional RNN then decoded into genre labels. The CNN-based attention mechanism is used to find the focus points.

We considered the last paper implementing the attention mechanism as an extension and tried to tune their models using the abalation study to find the better results. This work aims to provide a comparative study between different deep learning based architectures and the traditional machine learning classifiers that need to be trained with hand-crafted features.

METHOD

Our approach to MGC builds upon Feng et al.'s [6] proposed parallel CNN and bidirectional RNN hybrid architecture (PRCNN), by applying it to the Free Music Archive (FMA) dataset, instead of GTZAN. The key insight behind such an approach is that the convolutional neural network focuses on extracting spatial features from the input spectrogram data, whereas the recurrent neural network focuses on temporal features of the spectrogram. The results of the two networks, run in parallel, are then combined together into one representation. Finally, we use a softmax function for the multi-class classification problem of assigning genre labels.

In the following sections, we first describe the FMA dataset, and contrast it to GTZAN and other popular music datasets for MGC. Then, we go into detail about the hybrid architecture we used for the CNN and RNN components. Finally, we touch upon our attempts at extending our results by using attention-based mechanisms to focus the model on key areas of the spectrogram for MGC.

Music Datasets

Many different music datasets have become available in recent years, in response to greater demand for data that can be used to train deep learning models for classification or generation of music. Some of the popular music datasets that are frequently used include Free Music Archive (FMA) [2], GTZAN [12], Extended Ballroom [10], and RWC [7].

GTZAN

In Feng et al.'s work, the GTZAN dataset was used, which was assembled in 2002. The dataset consists of 1000 music excerpts of 30 seconds each. The dataset contains songs of 10 different genre categories: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae and Rock. There are 100 songs per genre. Because of the limited size of this dataset, the authors decided to split up the 30-second track into 5-second segments, to effectively increase the size of their train/test data by a factor of 6 and prevent overfitting [6]. However, despite the popularity of GTZAN as a dataset for MGC, an analysis was done on GTZAN which has revealed issues with its integrity, including: correctness of labels and the presence of duplicates [12]. Overall, 72 tracks were from the same recording and 160 tracks were mislabeled.

Free Music Archive (FMA)

On the other hand, the Free Music Archive (FMA) [2] has emerged in 2017 as a promising new dataset. FMA is a freely-available collection of audio features and metadata for about a million contemporary music tracks. The FMA dataset provides 917 GiB and 343 days of Creative Commons-licensed audio from 106,574 tracks, 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres. In FMA, each track is encoded as an .mp3 file with a sampling rate of 44,100 Hz and a bit rate of 320 kbit/s (263 kbit/s on average), in stereo. Moreover, the dataset also contains 518 pre-computed spectrogram features for each track. The features are computed over spectrogram windows of 2048 samples and 512 as the hop-length.

In Table 1, we provide a more detailed breakdown of FMA compared to GTZAN, as well as a few other popular music datasets for MGC.

Dataset	*tracks*	*genre*	*artists*	*published*
FMA-small [2]	**8,000**	**8**	-	**2017**
FMA [2]	106,674	161	16,341	2017
GTZAN [12]	1,000	10	300	2002
EBallroom [10]	4180	12	-	2016
RWC [7]	465	3	-	2001

Table 1. Comparison of popular MGC datasets, alongside our final chosen dataset: *FMA-small*, a subset of FMA.

Our Chosen Dataset

We decided to go with the more up-to-date FMA dataset due to its correctness, availability, scale, and rich metadata, especially when compared to GTZAN. Due to memory and computational limitations, the 'small' version of the FMA dataset (*FMA-small*) was used for this project. As a subset of the full data, it contains 8,000 30-second tracks and 8 balanced genres (1,000 tracks per genre), culminating in a total size of 7.2 GiB. The eight labeled genres are: *Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop, and Rock.*

We split our train-valid-test data based on an 8-1-1 ratio, leading to 6400 tracks for training and 800 tracks each for validation and testing. Thus, each of the eight genres has 800 training tracks, 100 validation tracks, and 100 testing tracks.

Spectrogram Generation

As input to the PRCNN model, we first converted the audio files into spectrograms. In Feng et al.'s approach, they used a regular spectrogram - the result of a short term Fourier transform (STFT) across the audio signal. We diverge from their work by choosing to use a log-mel spectrogram, which converts the regular spectrogram into the mel scale and also scales it by a log function. The mel scale makes the distance between frequencies evenly 'spaced', in terms of how it is perceived by a human ear. In other words, this makes the input to the

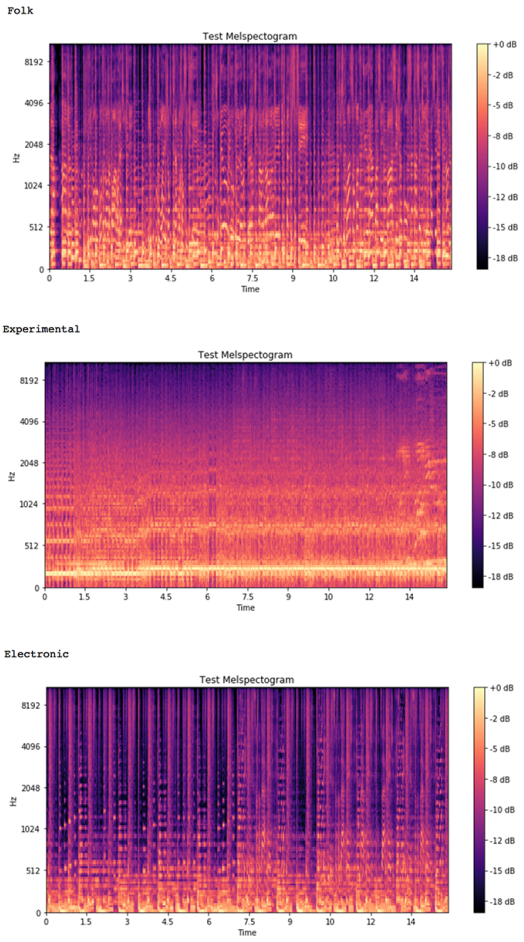


Figure 1. Generated mel spectrograms for three of the eight genres (from top to bottom: Folk, Experimental, Electronic) in the *FMA-small* dataset. Each genre has unique features that demonstrate the feasibility of the model's ability to differentiate between genres.

model more similar to something that a human would better be able to understand, which should improve performance. The log scaling converts the mel scale values into decibels, also to better match how humans perceive loudness. We generated a few choice mel spectrograms from 3 out of the 8 genres to better understand what the model 'sees' with regards to each genre label (see Figure 1). After this pre-processing stage, the log-mel spectrogram inputs to the model have a shape of (640, 128), the result of a STFT with 2048 as the window length, 512 as the hop length, and 128 as the number of mels.

CNN Component

Typically, convolutional neural networks (CNNs) star in image-based tasks, as the key elements in CNNs, the convolutional layers, are responsible for convolving images with filters or kernels, which are learned by the network. In audio-based tasks, the audio file is converted to a spectrogram image, which serves as the input for CNNs. By learning the spatial features of the spectrogram, the model is able to use the differences between representative spectrograms of each genre to distinguish between them.

The CNN component of Feng et al.'s [6] parallel architecture consists of 5 2D-convolutional layers, which are each followed by a 2D max-pooling layer. The ReLU activation function is used for each convolutional layer. All 5 layers have a kernel size of (3,1). For the first 3 layers, we used a pooling size of (2,2), whereas for the final 2 layers use a size of (4,4). Feng reports that this choice was made to extract more robust representations. Finally, the filter sizes are 16, 32, 64, 64, and 64 for each convolutional layer, starting from the first to last. In Feng's reporting of their CNN model, the fourth layer had a filter size of 128, which we felt was strange - in our early tests, we achieved better results with the filter sizes described above. The output of all convolutional layers is then flattened into a vector of (1,256).

RNN Component

For the RNN component, we take advantage of the utility of recurrent neural networks to extract temporal features. The intention of this block is to serve as a supplement to the CNN. Specifically, Feng et al. [6] use bidirectional Gated Recurrent Units (GRUs). The bidirectionality is motivated by being able to use both past and future information from the audio data, as there are no sequential constraints of only being able to see past data.

The RNN component begins with applying a 2D max-pooling layer of size (2,1) onto the same log-mel spectrogram data that the CNN component sees, as it is carried out in parallel. After pooling, the result is sent to a bidirectional GRU layer with 64 memory units, resulting in an output of (1,128).

With both the parallel outputs of the CNN and RNN components, we then concatenate both vectors, leading to a shape of (1,384). Finally, the model is wrapped up with a dense layer that uses softmax as the activation function.

Attention

In this section, we report on some of our experiments with adding an attention-based mechanism to the existing PRCNN model. This foray into attention was inspired by Yu et al.'s work [15], which is also largely an extension on Feng et al.'s work. By incorporating attention, we assign each temporal step a different weight, based on calculated attention scores. Intuitively, it is akin to looking at specific parts of the spectrogram/audio data with more emphasis than other parts. It learns which parts are important to look at (put 'attention' on) through a CNN-based model.

In our approach, we simply adopt the existing CNN component, as described above, but use the output for calculating attention scores instead. The attention scores are multiplied with the result of the RNN component, and then summed together, forming a weighted sum based on attention values. In essence, we follow the parallelized CNN-based attention model, as described by Yu et al [15], which was found to have the best accuracy scores compared to alternatives like serial attention and linear-based attention scores.

Implementation Details

Our implementation was primarily in Python 3 on a local Jupyter notebook. For the initial audio processing and spectro-

gram generation, we used Librosa. For the machine learning components of this project, we used Keras with a TensorFlow backend.

Initially, for training, it took about 20 hours to fully train the model after 50 epochs on our computers. This greatly limited the amount of experiments with the model that we could try, and made it difficult to fully implement the attention mechanism. We eventually managed to get access to GPUs through Compute Canada, which significantly sped up training time to about 15 minutes for the same number of epochs. We used the RMSProp optimizer from Keras with a learning rate of 0.0005 for training our model.

RESULTS

In this section, we report the key results from running our parallel RNN and CNN model, as well as our attempts at incorporating attention. For the parallel architecture, the best validation accuracy score that we obtained was 45%. The breakdown of the generated classification report from the test data is in Figure 2, which shows how the model fared on each of the genre labels.

	precision	recall	f1-score	support
Electronic	0.57	0.56	0.57	100
Experimental	0.26	0.37	0.30	100
Folk	0.28	0.34	0.31	100
Hip-Hop	0.59	0.87	0.70	100
Instrumental	0.47	0.32	0.38	100
International	0.54	0.41	0.47	100
Pop	0.38	0.21	0.27	100
Rock	0.58	0.51	0.54	100
accuracy			0.45	800
macro avg	0.46	0.45	0.44	800
weighted avg	0.46	0.45	0.44	800

Figure 2. The generated classification report from Keras for the PRCNN model, which has a test accuracy score of 45%.

In contrast, with an attention-extended approach, we obtained a validation accuracy score of 41.625%. The breakdown of this approach is in Figure 3. As we had limited time to experiment with and finalize our attention-extended approach, we are not completely surprised at the lower scores compared to the parallel approach. We discuss both results in further detail in the Discussion section below.

	precision	recall	f1-score	support
Electronic	0.49	0.62	0.55	100
Experimental	0.32	0.12	0.18	100
Folk	0.18	0.16	0.17	100
Hip-Hop	0.74	0.68	0.71	100
Instrumental	0.33	0.56	0.42	100
International	0.38	0.42	0.40	100
Pop	0.21	0.21	0.21	100
Rock	0.53	0.40	0.46	100
accuracy			0.40	800
macro avg	0.40	0.40	0.39	800
weighted avg	0.40	0.40	0.39	800

Figure 3. The generated classification report from Keras for the attention model, which has a validation accuracy score of 41.625%.

Ablation Studies

We performed several ablation studies to better understand how changes to the model might affect performance. First, we report on how each of the individual components of the parallel RNN and CNN architecture fare compared to the combined approach by and without attention. Then, we try tweaking parts of the individual components, such as changing the number of convolution layers in the CNN and experimenting with different pooling sizes. We used different pool sized of (2,2), (4,4), (2,1), and (4,2). We evaluated the results both based on the accuracy, F1-score, and recall. The final results are shown in Table 2. The architecture of our model using attention can be seen in Figure 4. The hyper-parameters we used to tweak the results are the combination of filter-sizes, kernel size, drop-out probabilities and the layer activations. The best result we get both in training and test set is derived by using 5 CNN layers with the filter sizes of 16, 32, 64, 64,64 with the pooling layers of (2,2), (4,4) and (4,2) and the kernel size of (3,1). We also trained a model using the attention in the architecture which leads to getting comparable result of 0.41625 for the validation score but much less test accuracy of 0.2625 compared to our best model performance.

Of note are that these scores, and those in Table 2 are different from that of reported in the Results section above; this is because we conducted our ablation studies entirely on the GPU. We ran into compatibility and versioning issues when working on the GPU compared to our local machines, which resulted in lower accuracy scores on the GPU, despite the same model. With more time, we hope to be able to further investigate these issues, but our lack of experience and control over our GPU access limited what we could do with it. Despite this setback, it is still interesting to see how the 5 layer only CNN model performed the best. This indicates that the CNN is the main contributor to performance in the parallel CNN-RNN model, and that the RNN does not improve performance in this case.

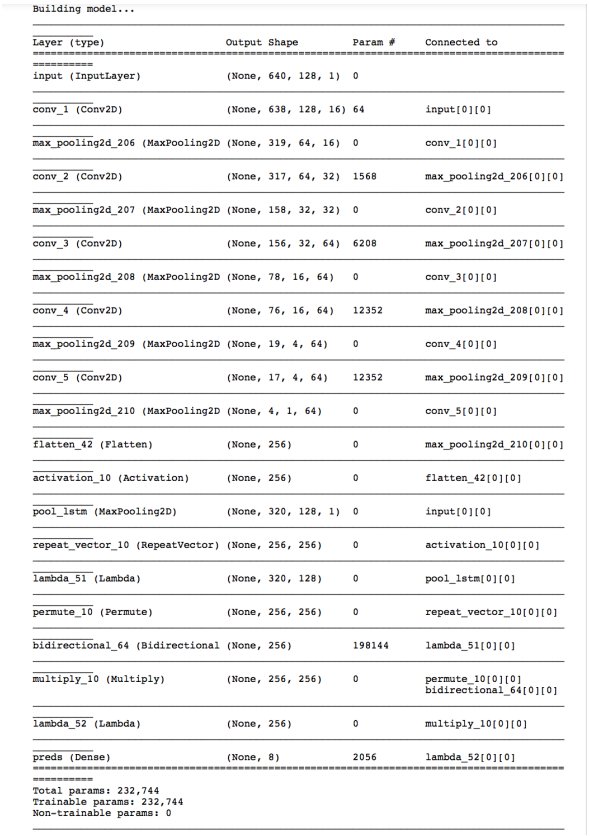
DISCUSSION

The results of the parallel RNN and CNN approach on the FMA dataset were fairly lackluster, especially when compared to the experimental results from Feng et al.'s work with the same architecture on the GTZAN dataset. Compared to our results, they boasted accuracy scores of up to 92% for the same MGC task [6]. Although we report much lower scores, this discrepancy in replicability is interesting, and points to several potential possibilites, which we will discuss below.

First, the differences in dataset are likely to play a large role in the differences in accuracy scores. It could be that the FMA dataset is more challenging to work with, and more representative of a real-world MGC problem. It is also important to keep in mind the flaws of GTZAN, as discussed extensively earlier in the Method section, when assessing accuracy scores on that particular dataset. In our background research, we found some evidence of the challenge and relevance of FMA and the problem of MGC. One of the WWW (Web Conference) 2018 challenges was Music Genre Classification, with the FMA dataset [5]. The winning team had an accuracy score of 63% with the full FMA dataset (more data, but also more genres). This result gives our results a target to compare against, and grounds it more realistically in the realm of accuracy scores for the FMA dataset, instead of GTZAN. Also, as we only used

Model	*Validation Accuracy*	*Test Accuracy*
Model using RNN	40.375%	29.5%
Model using 5 layers CNN	51.25%	44.125%
Model using 3 layers CNN	39.4%	26%
Model using CNN and 1 layer RNN	45%	36.375%
Model using CNN and two layer RNN	46.25%	35.5%
Model using CNN and two layer RNN and the attention	41.625%	26.25%

Table 2. Evaluation of trained models from our ablation studies. Accuracy scores were calculated on the GPU, which surprisingly had different scores than on our local machines, despite being the same model.



```
Building model...
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input (InputLayer)           (None, 640, 128, 1)       0
_________________________________________________________________
conv_1 (Conv2D)              (None, 638, 128, 16)      64
_________________________________________________________________
max_pooling2d_206 (MaxPooling2D (None, 319, 64, 16)    0
_________________________________________________________________
conv_2 (Conv2D)              (None, 317, 64, 32)       1568
_________________________________________________________________
max_pooling2d_207 (MaxPooling2D (None, 158, 32, 32)    0
_________________________________________________________________
conv_3 (Conv2D)              (None, 156, 32, 64)       6208
_________________________________________________________________
max_pooling2d_208 (MaxPooling2D (None, 78, 16, 64)     0
_________________________________________________________________
conv_4 (Conv2D)              (None, 76, 16, 64)        12352
_________________________________________________________________
max_pooling2d_209 (MaxPooling2D (None, 19, 4, 64)      0
_________________________________________________________________
conv_5 (Conv2D)              (None, 17, 4, 64)         12352
_________________________________________________________________
max_pooling2d_210 (MaxPooling2D (None, 4, 1, 64)       0
_________________________________________________________________
flatten_42 (Flatten)         (None, 256)               0
_________________________________________________________________
activation_10 (Activation)   (None, 256)               0
_________________________________________________________________
pool_lstm (MaxPooling2D)     (None, 320, 128, 1)       0
_________________________________________________________________
repeat_vector_10 (RepeatVector) (None, 256, 256)       0
_________________________________________________________________
lambda_51 (Lambda)           (None, 320, 128)          0
_________________________________________________________________
permute_10 (Permute)         (None, 256, 256)          0
_________________________________________________________________
bidirectional_64 (Bidirectional (None, 256)            198144
_________________________________________________________________
multiply_10 (Multiply)       (None, 256, 256)          0
_________________________________________________________________
lambda_52 (Lambda)           (None, 256)               0
_________________________________________________________________
preds (Dense)                (None, 8)                 2056
=================================================================
Total params: 232,744
Trainable params: 232,744
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. Output of the built architecture of our attention-extended model.

the smaller *FMA-small* dataset, we might not have had enough data for our deep learning models to be performant. Adding more data, or adopting the approach to split up data into segments, as Feng et al. did, might be critical to improving our accuracy scores.

Second, these results could also call into criticism the reporting and replicability of Feng et al.'s work [6]. For our team, a major challenge was in simply replicating the described PRCNN architecture, as key details were missing, ambiguous, or incorrect. One important limitation of our contribution in this project is that we did not train and test our replicated model on the GTZAN dataset, to see if we get similar accuracy scores that they reported. By doing so, we could have been more confident that we correctly implemented and replicated their PRCNN architecture.

Finally, this also leads us to a criticism of our own work: it may be that the lower scores are a result of an imperfect, or naive implementation. Given more time, and earlier access to GPUs, we could have experimented with different CNN/RNN architectures or with different kernel sizes, which was fixed at (3,1) for all convolutional layers. This is especially true of our attention-extended approach, as it should not have decreased the accuracy scores, unless it has somehow overfit. We also faced some difficulties with incompatible TensorFlow versions when transitioning between training models on the GPU and when testing them on our local machine, which resulted in different accuracy scores, despite the same architecture. Part of this could be attributed to our inexperience with working with GPUs and with complex deep learning models.

As a concluding note, we discuss the genres in which accuracy was much lower than the others. Interestingly, it appears that the model has difficulty in classifying the Experimental, Folk, Instrumental, and Pop genres, as they each have an F1-score of lower than 0.4. To improve performance, it may help to have

more data in these genres. It may also be worth it to conduct a more detailed analysis on the spectrograms of these genres, and to see if there are enough distinct features for each in order to classify them appropriately using just spectrogram data. If not, alternative features or data types might be necessary to supplement the model.

CONCLUSION

We have replicated and extended the results of Feng's parallel recurrent and convolutional neural network (PRCNN) architecture to the newer FMA dataset, while incorporating slight modifications to the chosen hyperparameters and adopting log-mel spectrograms instead of regular ones. In addition, we took a step at extending their work with attention-based mechanisms. We achieved test accuracy scores of 45% for the PRCNN approach and 40% for the attention-based approach, which still needs polishing. Future work includes refining both models - in particular, the attention-based approach. In addition, experimenting with other data sources to supplement spectrogram data may be critical, as not all genres might be differentiable on the basis of their spectrogram alone. Our initial motivation for looking at MGC was to automatically include lyrics as a feature by extracting them directly from a song, and that still remains as a potential promising future direction.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Robert Xiao and our peers for their valuable feedback and suggestions on this project through our in-class paper and demo presentations.

REFERENCES

[1] 2019. Every Noise at Once. (2019). http://everynoise.com/.

[2] Kirell Benzi, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2016. FMA: A Dataset For Music Analysis. *CoRR* abs/1612.01840 (2016). http://arxiv.org/abs/1612.01840

[3] k. Choi, G. Fazekas, M. Sandler, and k. Cho. 2016. Convolutional recurrent neural networks for music classification. *rXiv preprint arXiv:1609.04243* (2016).

[4] Oliveira L.S. Silla C.N. Costa, Y.M. 2016. An evaluation of convolutional neural networks for music classification using spectrograms. (2016), 28–38.

[5] Michaël Defferrard, Sharada P. Mohanty, Sean F. Carroll, and Marcel Salathé. 2018. Learning to Recognize Musical Genre from Audio. In *WWW '18 Companion: The 2018 Web Conference Companion*. https://arxiv.org/abs/1803.05337

[6] Lin Feng, Sheng-lan Liu, and Jianing Yao. 2017. Music Genre Classification with Paralleling Recurrent Convolutional Neural Network. *CoRR* abs/1712.08370 (2017). http://arxiv.org/abs/1712.08370

[7] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. 2002. RWC Music Database: Popular, Classical, and Jazz Music Databases. In *In Proc. 3rd International Conference on Music Information Retrieval*. 287–288.

[8] Maura Johnston. 2018. How Spotify Discovers the Genres of Tomorrow. (2018). https://artists.spotify.com/blog/how-spotify-discovers-the-genres-of-tomorrow.

[9] Dylan Freedman Aren Jansen Wade Lawrence R Channing Moore Manoj Plakal Jort F Gemmeke, Daniel PW Ellis and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. *In Acoustics, Speech and Signal Processing (ICASSP)* (2017).

[10] Ugo Marchand and Geoffroy Peeters. 2016. The extended ballroom dataset. (2016).

[11] AB Mutiara, R Refianti, and NRA Mukarromah. 2016. Musical Genre Classification Using SVM and Audio Features. *TELKOMNIKA Telecommunication, Computing, Electronics and Control 14(3)* (2016).

[12] Bob L. Sturm. 2012. An Analysis of the GTZAN Music Genre Dataset. In *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies (MIRUM '12)*. ACM, New York, NY, USA, 7–12. DOI: http://dx.doi.org/10.1145/2390848.2390851

[13] G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* (2002), 10(5):293–302.

[14] Shenglan Liu Hong Qiao Yang Liu Yang Yu, Sen Luo and Lin Feng. 2019. Deep attention based music genre classification. *Neurocomputing* (2019).

[15] Yang Yu, Sen Luo, Shenglan Liu, Hong Qiao, Yang Liu, and Lin Feng. 2020. Deep attention based music genre classification. *Neurocomputing* 372 (2020), 84–91.