CMPE/CISC 452: PROJECT REPORT
MUSIC GENRE CLASSIFICATION USING DEEP LEARNING

Qiyuan Cai, 10177331, astral.cai@queensu.ca
Hong Yi Xiong, 10176644, hong.yi.xiong@queensu.ca
Ziping Li, 10178011, 14zl63@queensu.ca
Yishan Li, 10182827, 14yl110@queensu.ca

I. CONTRIBUTIONS

The table below shows individual contributions to implementing the project. Each member wrote their corresponding sections in the report (Hong Yi wrote about the selection and pre-processing of data. Ziping, Yishan, and Qiyuan wrote about the implementation and performance results of the models they each implemented). The Problem Description was mostly taken from the proposal. The Discussion and Future Improvements sections were written collectively. Qiyuan Cai was in charge of combining all sections and editing the report.

TABLE I
INDIVIDUAL CONTRIBUTIONS BY GROUP MEMBERS TO IMPLEMENTING THE PROJECT

Hong Yi Xiong	Data selection, acquisition and pre-processing
Ziping Li	Transfer learning model with pretrained network (VGG-16)
Yishan Li	Testing different custom CNN structures
Qiyuan Cai	RNN model, and parallel RNN+CNN model (with attention)

II. PROBLEM DESCRIPTION

Music has always been a huge part of people's lives. With the growth of the internet over the past decade, usage of online music streaming services and corresponding applications increased visibly. In the meantime, sharing music online has become much easier, which leads to an exploding size of music databases online. As a result, music classification becomes crucial in sorting music databases and developing recommending systems for music. As databases grow larger, classifying each song manually becomes unpractical. To solve this problem, automatic classification of music based on genres has become a popular field of research.

Over the past decade, different algorithms have been developed for music genre classification. In recent years, deep neural networks such as convolutional neural networks (CNN) and recurrent neural networks (RNN) are gaining popularity [11]. They have been shown to perform very well in music information retrieval tasks. A CNN can be trained on spectrogram representations of audio clips, and Bahuleyan [1] has shown that this network out-performs generic classifiers trained on conventional hand-crafted features. Researchers such as Jeong and Lee [7] and Li et al. [8] has also shown promising results using CNNs and RNNs for feature extraction.

In this project, we studied different deep neural network structures for music genre classification. We experimented with different CNN-based and RNN-based models, as well as a hybrid model which combines a CNN and an RNN, first proposed by Feng et al. [4]. We incorporated various techniques proposed by Yu et al. [14] and Choi et al. [3], and compared their effects on the classification accuracy.

III. DATA

We will use two datasets in this study, Tzanetakis' GTZAN dataset [13] and the Music Audio Benchmark Data Set [5] from the Garageband. We decided to choose 2 different datasets because we wanted to test how well our algorithms can generalize on different datasets. It has been shown that the performance of a music genre classification model is closely correlated with the dataset used to train the network [9].

The GTZAN dataset consists of 1000 audio tracks, each 30 seconds long. It includes 100 tracks for each of the 10 genres. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. The genres of music that this dataset contains are rock, hip-hop, pop, jazz, blues, classical, country disco, metal and reggae. This data set is used as the benchmark for most music genre classification studies.

Garageband is a website that allows artists to upload their music and offer it for free download. This dataset contains 1886 songs all being encoded in .mp3 format. Each song is associated with a 10-second audio sample drawn from a random position of the corresponding song. Audio samples have a sampling rate of 44100 Hz and a bitrate of 128 mbit/s. The genres of music that this dataset contains are rock, hip hop, pop, jazz, blues, alternative, electronic, folk/country and funk/soul.

A. Data Selection

For the scope of this project, we trained and tested our models with a subset of the two datasets. We used audio clips of 5 different genres, including rock, hip hop, pop, jazz and blues. Most studies have shown that the classification accuracy varies significantly among different genres. To properly compare the performance of our models on these two datasets, we chose these five genres which are contained in both datasets mentioned above.

B. Data Preprocessing

The datasets consists of raw audio files, which are not directly useful for feature extraction. It is easy to imagine that the frequency domain representation of audio clips are more representative of their characteristics. As a result, most studies on music information retrieval used the spectrogram representation of the raw samples. A spectrogram is a visual representation of the input signal's frequency spectrum [14]. The spectrogram of the audio samples can be calculated by the squared magnitude of its short-time Fourier transform (STFT).

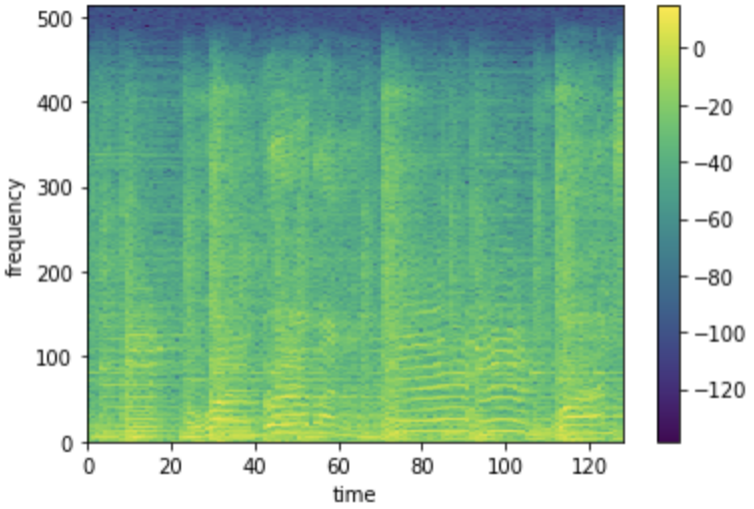

Fig. 1. An example of a spectrogram converted from a 3-second audio clip using short-time Fourier transform (STFT).

The audio samples in the two datasets have different sample rates. Specifically, the Benchmark dataset was sampled at 22050Hz, and the GTZAN dataset was sampled at 44100Hz. To ensure consistency, the audios from the GTZAN dataset were re-sampled to 22050Hz. Next, the audio files were divided into 3-second clips. Each audio file in the Benchmark dataset was 10 seconds long, which was divided into three samples. The GTZAN dataset on the other hand consists of 30-second audio samples, each divided into ten samples.

The STFT involves several parameters. The most important one is the choice of a sampling window. The Hanning window and Tukey window are two popular choices. In this project, the Hanning window was chosen because it yielded the better result. The window length is another important parameter. A longer window yields better resolution in frequency, but lower temporal resolution, and vice versa [2].

For the purpose of music genre classification, we decided that frequency domain features were more important. Therefore, we chose a relatively longer window size of 1024. Different window sizes such as 512 and 216 were also tested, but the 1024 window yielded the best results, which confirmed our hypothesis. The overlapping ratio was chosen to be 50%. Higher values would be more computationally expensive, but lower values would give lower quality. In the end, 50% worked best with the Hanning window. The final sample is a 2-dimensional matrix of size 513×128, where 513 is the number of frequency divisions, and 128 is the number of time steps.

IV. Implementation

All models were built and trained using the `Keras` library and `TensorFlow`, in the Google Colaboratory environment (Python 3 Google Compute Engine Backend, runtime type: GPU, RAM: 25.51GB, disk space: 358.27GB). Data pre-processing was completed on a laptop (Windows 10, Intel CPU i7 2.4GHz, RAM: 16.0GB). The spectrograms converted from raw audio clips were then uploaded onto the Google Drive so that they could be accessed from Google Colab.

All our models consists of a feature extractor followed by a fully connected classifier. The classifier consists of one hidden layer and one output layer of five nodes. The activation function for the hidden layer is chosen to be ReLU, and the output layer uses softmax activation. The loss function used is categorical cross-entropy, defined below, where y is the actual probability for each category, and p is the prediction vector of the network.

$$\mathcal{L} = -\sum_{c=1}^{M} y_{o,c} * \log p_{o,c} \tag{1}$$

The number of hidden units in the classifier depends on the output dimensions of the feature extractor. We tested several different structures, including two convolutional neural network structures, a recurrent neural network model, and a hybrid network which combines the CNN and the RNN.

A. CNN-based Model

One of the most common approaches to music genre classification is to treat it as an image pattern recognition problem using spectrogram representations of audio samples. Several studies have shown that the convolutional neural network (CNN) performs very well for this task [1, 3, 8]. The standard approach is to connect the output of the convolutional layers to a fully-connected softmax classifier. We tested two different CNN models: a custom structure we built ourselves and a pre-trained network.

Our own convolutional neural network structure consists of 5 convolutional layers, each followed by a max-pooling layer to reduce the dimensionality of the feature map obtained from the convolutional steps (also known as down-sampling). We chose ReLU to be the activation function for all convolutional layers. This is the convention because the ReLU function does not saturate at high activation values, which kills of the gradient, as is found to converge faster than other activation functions.

The kernel size for all convolutional layers is 3×3, and the number of channels in each layer are 32, 32, 64, 128, 64, from the input layer to the output layer of the convolutional block, respectively. This structure is seen in several different studies, such as one conducted by Yu et al. [14] and by Feng et al. [4]. The output of the last convolutional layer is flattened and fed into a densely connected layer with 64 units, followed by a softmax output layer of 5 nodes.

B. Pre-trained CNN

The pre-trained network we chose to use was VGG-16, as suggested by Bahuleyan [1]. The VGG-16 was trained by team VGG from Oxford University as a submission to the Large Scale Visual Recognition Challenge in 2014 [12]. The original VGG-16 is a convolutional neural network pre-trained using the ImageNet dataset, which contains over 15 million labeled real-world images. The VGG-16 network consists of a stack of convolutional layers, followed by two fully connected layers. Figure 2 below illustrates the structure of the VGG-16 model.

We replaced the original pre-trained fully connected layer and softmax classifier on top of the convolutional layers with our own fully connected hidden layer, and a softmax output layer of 5 nodes. Different number of hidden units were tested, and 17 produced relatively better results. The fine-tuning option on the convolution layers was turned on, and the model was then trained end-to-end with the spectrograms.

The VGG-16 model takes images of size 224×224 of 3 color channels. We saved the pseudocolor plot representations of the spectrograms (obtained using the `matplotlib.pyplot.pcolormesh` function) like the one shown in Figure 1 above, with all axis labels and legends removed. Then we useed the `matplotlib.pyplot` library to resize the extracted images to the desired size of 224×224. Finally, we used the `TFRecordWriter` of `TensorFlow` to convert these images to `tfrecord` files, which takes shorter time to load compared to the raw image files. This accelerates the training process significantly.

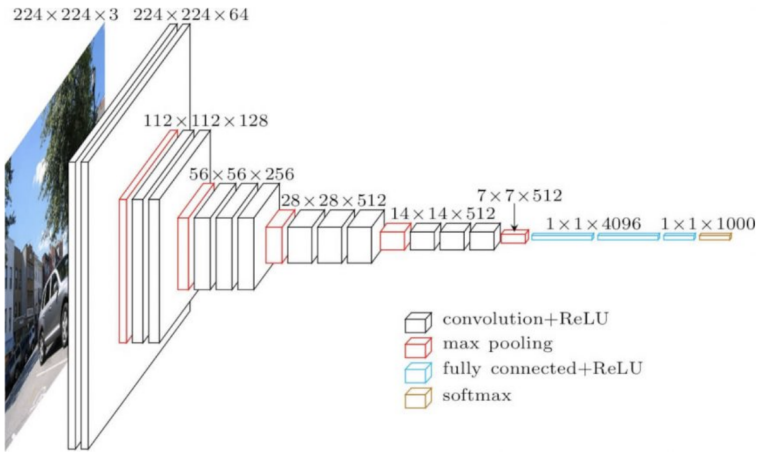

Fig. 2. Network structure of the VGG-16 pre-trained model [12]

C. RNN-based Models

Another popular choice of deep neural networks for music genre classification is recurrent neural networks (RNN). It is able to capture the temporal information and sequential relationship within the input data, which is also a crucial part of identifying features such as music structures or recurrent harmonies [4]. Most related studies use a bi-directional RNN, which consists of a forward layer and a backward layer. The bi-directional RNN is able to model backward dependence in the input data. The RNN structure we chose was the gated recurrent unit (GRU). It is a simplified version of the long short-term memory algorithm. This algorithm allows the network to adaptively control how much the previous states affect the present. Figure 3 below illustrates the structure of a bi-directional gated recurrent unit (GRU) block.

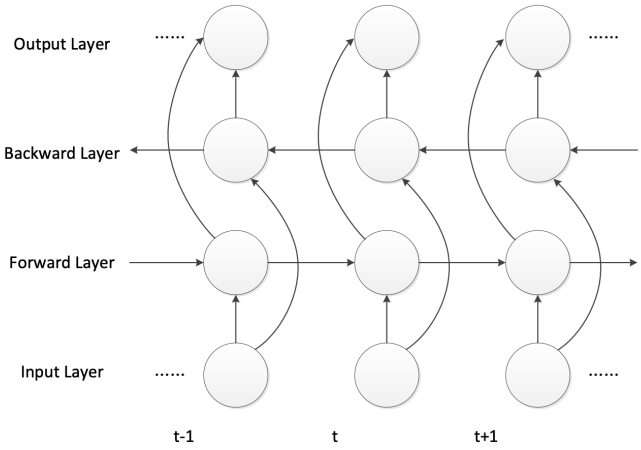

Fig. 3. Structure of a single BGRU block [4]

Our model consists of two stacked bi-directional GRUs, each with 256 units. The input to the model is a sequence of spectrum vectors extracted from the spectrograms. Each audio sample is treated as a series of 128 spectrum vectors with a length of 513. We chose to have each layer return the entire sequence of inputs. The output dimension of each BGRU layer turns out to be 128×512, where 512 comes from concatenating the forward and backward layers in each BGRU block. The time average of the final sequence of outputs from the second BGRU layer is computed, and the result is an output vector of 512 elements. This vector is fed into a fully-connected layer of 64 nodes, followed by a softmax output layer of 5 nodes that serves as a classifier.

D. Attention Mechanism and the Parallel Network

We also explored different techniques to enhance the performance of our models. One of them is the attention mechanism. The intuition is that, when humans listen to music, we pay more attention to part of the music than to the rest. As a result, information at some temporal steps should be given more importance. This is achieved with an attention mechanism, first proposed by Yu et al. [14]. The attention mechanism is in short a network that produces a score vector, where each element represent the attention given to its corresponding time step in the sequence of data. We used the attention mechanism on the BGRU network mentioned in the section above. Given the sequential output of the second BGRU block, instead of a simple time average, we take a weighted average of the vectors at each temporal step, where the weights are the output vector of the attention network. This structure is illustrated in Figure 4 below.

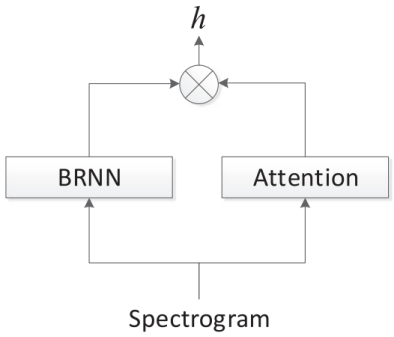

Fig. 4. The structure of an BGRU network with attention mechanism [14]

As suggested by Yu et al. [14], the convolutional neural network performs very well as an attention network. We chose the same CNN structure specified in the previous section, and added a dense layer with 128 units on top of the convolutional layers, with a softmax activation function, so that the output is normalized. This output vector is treated as the attention scores for each time step. We take the dot product of the BGRU output and the attention score vector, which produces a 512 dimensional output vector. This is finally fed into a fully connected classifier with one hidden layer of 64 nodes, same as before.

Finally, we experimented with a hybrid network. We took the BGRU network with attention incorporated, and the convolutional layers specified before. We added a dense layer with 512 units on top of the convolutional layers so that the output dimension agrees with the BGRU network. We then concatenated the output vectors of the BGRU network and the CNN, feeding the output into a fully connected classifier with one hidden layer of 80 units, and a softmax output layer of 5 nodes, same as before.

E. Ways to Prevent Over-fitting

We found that our networks tend to have a large number of parameters. Given the small size of our training data, our networks were very prone to over-fitting. As a result, we used several regularization techniques in our models. The first approach was to reduce the number of parameters if possible. The GBRU network proposed by Yu et al. [14] concatenates the output of both BGRU layers, which doubles the number of parameters. To decrease the complexity of the model, we chose to only use the output of the final BGRU layer, which improved the classification accuracy by around 2%. Another approach is to use validation data during training. The model was set to stop training and revert to the weights which produced the best results if the validation accuracy does not improve for 3 consecutive epochs. We chose to use 20% of the training samples as validation data.

We also added dropouts to the classifier, where some neurons are ”shut off” for every iteration. This prevents the network from depending too much on the outputs of a small subset of neurons. The dropout rate was set at 0.3. We found that a lower dropout rate would increase the network’s tendency to over-fit, whereas a higher rate would make the network too hard to train. We also used L2 regularization, which is an adjustment on the loss function so that it penalizes excessively high weights.

V. RESULTS

We used 80% of our dataset for training, and the rest for testing. This results in 3618 training samples, and 905 test samples with the GZTAN dataset, and a 3261-816 split betweeen training and testing with the Benchmark dataset. At first, we tried to train with a mixture of the two datasets. However, the result performance was poor. It is very likely that the two datasets have different labelling standards, which confuses the model. Therefore, we tested our models with these two datasets separately and measured the classification accuracy, macro-average F1-scores, and AUC for each model. The results are summarized in the tables below.

TABLE II
EXPERIMENTAL RESULTS ON GTZAN

Network	**Accuracy**	**F1-score**	**AUC**
CNN	0.83	0.84	0.97
BGRU	0.84	0.84	0.97
BGRU+Attention	0.88	0.88	0.98
Parallel BGRU+CNN	**0.90**	**0.90**	**0.98**

TABLE III
EXPERIMENTAL RESULTS ON BENCHMARK

Network	**Accuracy**	**F1-score**	**AUC**
CNN	0.68	0.51	0.91
BGRU	0.70	0.60	0.90
BGRU+Attention	0.68	0.52	0.91
Parallel BGRU+CNN	**0.72**	**0.62**	**0.91**

The performance of the VGG-16 based network was not included in the tables above because the performance was poor and extremely inconsistent. In general, the loss of the VGG-16 based model was in a trend of reducing throughout the training process. However, it was not dropping in a stable manner. The loss was jumping up and down, even with the learn rate set to as low as 0.00006. After several epochs of training, the loss usually stabilizes around 1.6 to 1.7, the final training accuracy usually falls between 70% and 85%, but the testing accuracy goes as high as only 50% to 60%, and it often drops below 40%.

TABLE IV
CONFUSION MATRIX FOR THE BGRU+CNN MODEL ON GTZAN

Genres	Blues	Hiphop	Jazz	Pop	Rock
Blues	147	3	11	1	4
Hiphop	3	168	2	2	0
Jazz	1	1	190	4	7
Pop	1	0	6	169	13
Rock	7	3	16	10	136

TABLE V
PERFORMANCE REPORT OF THE BGRU+CNN MODEL ON GTZAN

Genres	precision	recall	f1-score	support
Blues	0.92	0.89	0.90	166
Hiphop	0.96	0.96	0.96	175
Jazz	0.84	0.94	0.89	203
Pop	0.91	0.89	0.90	189
Rock	0.85	0.79	0.82	172

To better understand the performances of our models with classifying different genres, we evaluated our networks on the GTZAN dataset in more detail, shown in the tables above. Table IV and Table V were evaluations on the parallel BGRU+CNN model. The performance of other models showed a similar pattern. We can see that

the model does most poorly recognizing rock music. This is consistent with the discoveries of several other studies. This result was somewhat expected. As Huang et al. [6] pointed out, a qualitative inspection of the spectrograms would show that the beats in rock spectrograms are usually less recognizable, which seems to be an important part of music genre recognition.

In general, we see that the models perform better with the GTZAN dataset than with the Benchmark dataset. This was partly because our models were tuned with the GTZAN dataset. This confirms what we learned in the literature, that the performance of a model is highly correlated with the specific dataset it was designed with. We see that the parallel BGRU+CNN network with attention mechanism incorporated performs best among all four models. The RNN networks generally performed better than the CNN-based models. It is also worth noting that the attention mechanism improved the performance of the BGRU on the GZTAN dataset. However, when testing with the Benchmark dataset, the result was the exact opposite. Even when testing with the GTZAN dataset, the performance improvement brought by the attention mechanism was not always significant.

VI. Discussion and Future Work

One possible reason behind the poor performance of the VGG-16 based model is that the model was originally trained to detect objects in an image, which did not transfer well to the music genre recognition task. The convolution filters in the VGG-16 network were trained to extract shapes and outlines instead of detailed features in a spectrogram. Another issue we faced with the VGG-16 model was that its required input size. By downsizing the original spectrograms to the required 224×224 input images, we lost half the frequency resolution, which is crucial for all music information retrieval tasks.

In most other studies in literature, CNN-based models generally perform better than RNN-based models. However, our experiment has shown the exact opposite. It is very likely that we have not designed the best structure for the task, and it is potentially valuable to keep exploring different CNN structures in the future. As shown in Table II above, the parallel BGRU+CNN model performed the best among all models, which suggests that the textures extracted by the CNN and the temporal features extracted by the RNN both contain information that helps the prediction of the models.

As mentioned above, the attention mechanism did not improve the performance of the BGRU as much as expected. This is worth investigating if given more time. Since the attention mechanism was implemented using a convolutional neural network, this issue might be related to the poor performance of our CNN-based models. Another potential area of improvement is to explore different network structures for the attention mechanism. Yu et al. [14] also suggested parallel linear attention and serial attention networks. It might be worth validating the performances of these structures.

Since we only used a subset of the datasets in our study. Our results are not directly comparable with the results of other studies. The current state-of-the-art prediction accuracy in music genre classification is 94.38% achieved by Panagakis et al. [10] on the ISMIR2004 dataset by modelling the auditory cortical representations of music recordings. The results shown in Table II and Table III were selected from multiple trials. The performance of the models in our experiment fluctuated on a trial-to-trial basis. This suggests that our models still cannot handle the complexity of the problem very well. However, we see that the AUC scores for our models were mostly close to 1, which suggests that our models did well in distinguishing between the genres. In the future, it might be beneficial to train and test with a larger dataset, which would help the network's ability to generalize. However, such datasets are not so easily found.

References

[1] Bahuleyan, H. (2018). Music genre classification using machine learning techniques.

[2] Boashash, B. (2015). *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press.

[3] Choi, K., Fazekas, G., and Sandler, M. (2016). Explaining deep convolutional neural networks on music classification.

[4] Feng, L., Liu, S., and Yao, J. (2017). Music genre classification with paralleling recurrent convolutional neural network. *CoRR*, abs/1712.08370.

[5] Homburg, H., Mierswa, I., Möller, B., Morik, K., and Wurst, M. (2005). A benchmark dataset for audio classification and clustering. In *Proc. 6th Int. Conf. Music Information Retrieval*, pages 528–531.

[6] Huang, D. A., Serafini, A. A., and Pugh, E. J. (2019). Music genre classification. unpublished.

[7] Jeong, I.-Y. and Lee, K. (2016). Learning temporal features using a deep neural network and its application to music genre classification. In *ISMIR*.

[8] Li, T. L., Chan, A. B., and Chun, A. H. (2010). Automatic musical pattern feature extraction using convolutional neural network. In *In Proc. IMECS*.

[9] Nanni, L., Costa, Y. M., Lumini, A., Kim, M. Y., and Baek, S. R. (2016). Combining visual and acoustic features for music genre classification. *Expert Systems with Applications*, 45:108 – 117.

[10] Panagakis, Y., Kotropoulos, C., Informatics, D. O., and Arce, G. R. (2009). Music genre classification using locality preserving nonnegative tensor factorization and sparse representations. In *In 10th International Society for Music Information Retrieval Conference (ISMIR*.

[11] Rajanna, A. R., Aryafar, K., Shokoufandeh, A., and Ptucha, R. W. (2015). Deep neural networks: A case study for music genre classification. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 655–660.

[12] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

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

[14] Yu, Y., Luo, S., Liu, S., Qiao, H., Liu, Y., and Feng, L. (2019). Deep attention based music genre classification. *Neurocomputing*.