CNN-RNN Model for Automated Music Genre Classification

Sikilammetla Sai Kiran
School of computer science and engineering
Lovely Professional University
Phagwara, Punjab, India

***Abstract*— Automated music genre classification is an essential task in audio analysis, supporting applications in personalized recommendations, music discovery, and content management. This study presents a hybrid Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) model developed to classify music genres based on audio features. This model uses the GTZAN Genre Classification dataset consisting of a thousand audio tracks spread over 10 genres; the CNN layers in this model filters spatial patterns from the audio spectrograms while the RNN layers the temporal features important for effective genre distinction. Select preprocessing were used such as feature scaling and visualization for the purpose of improving the performance of the models. The model reached the test accuracy of 87.29% and the test loss, 0.5285 pointing out the ability of the model in genre classification. This paper proves that CNN-RNN is efficient in audio classification and gives a general perspective on deep learning for automated music genre recognition system.**

Keywords—Music Genre Classification, Audio Analysis, CNN-RNN Model, Convolutional Neural Network, Recurrent Neural Network, Audio Feature Extraction.

1. Introduction

Classification of music genre is a important subfield of audio analysis used in personal recommendation systems, developing digital music libraries, and more relevant and automatic indexing of audio content. Due to the increasing number of music data available on the internet, it becomes very important to classify current and diverse music genres with high efficiency. More simple approaches used in other music genre classification problems such as hand engineered feature extraction and relatively simple learning algorithms may not be sufficient in capturing as much detail in these audio data because adaptation to the change that is constantly happening in the music industry require more sophisticated machine learning techniques.

The latest development in deep neural networks, especially in the CNN-RNN compound architecture, has made the processing of audio signals completely new. CNNs outperform in spatial representation learning from spectrogram representations of audio while RNNs are better suited to modelling temporal structure. Such synergy helps provide a much better analysis of the music, which in turn leads to better classification. Due to the ability of CNN-RNN models to directly learn the feature from the given audio data, the performance of music genre classification systems is improved with the help of such methods instead of conventional techniques.

Also, the existence of well-selected datasets these are the GTZAN Genre Classification dataset in particular have helped in training and testing deep learning mechanisms. This data set contains music of different varieties, covering many types of genres, which makes this data set useful for studying the peculiarities of the genre classification. When using the data present in this particular dataset, it is possible to derive the machine learning models that not only perform the task of music genre classification but also help in the enhancement of the understanding of fundamental and defining features of the said genres.

However some problems do persist with regards to music genre classification, where basic obstacles such as overlapping and variability in the quality of sound still exist. These factors can act negatively to increase the steps and difficulty of model training and possibly damages the reliability of the later classification. Therefore, solutions designed to tackle these difficulties like data augmentation, fine-tuning of models are inevitable for obtaining good performance.

In the present work, a new model of CNN-RNN has been introduced for automatic music genre classification. Our approach combines the capabilities of CNN and RNN, which enhances classification accuracy, and deals with typical issues in music genre classification. We are very careful on data preprocessing and model hyperparameters, and our ultimate goal is to provide useful information to the field of audio analysis and classification.

2. Literature Review

Content classification of music genre has received much attention in recent years because the use of technology in producing music has increased tremendously and classification technology is critical in categorizing music. Most previous studies on genre classification utilize text features such as auditory features, and prior works have used shallow learning techniques that fail to capture the subtle features of audio data. Therefore the trend is to use more complex machine learning models that would be able to extract the relevant features of the given data and increase the classification performance. The first papers demonstrated that rule-based methods are not so effective for MIDI classification because, often, they could not take into account the variability of different types of music and, due to this, the recognition's quality and speed were not high, while the rates of misclassification were high too [1].

Recently, two new approaches to DL has brought considerable improvement in genre classification: CNN and RNN. CNNs are also very useful in developing spatial hierarchy within the data, thus ideal for developing spectrogram representation of signals. Of the RNNs, LSTM has been found especially effective for analyzing sequential data, which is very important for modelling temporal structure in music [2]. Chen et al., in his study establish the applicability of CNNs for music genre classification on grounds of better accuracy rates as opposed to conventional techniques [3]. Also, integrating CNNs with RNNs proved to exploit both the spatial and the temporal characteristics of the used audio data, improving the performance of the model [4].

Still, one of the problems of music genre categorization is the question of signal quality variation and also overlaps between them. In response to these challenges, authors have considered data augmentation methods for synthesising modified versions of given audio samples to add variability of sounds. For example, Huang et al. (2020) utilized processes like pitch shifting and time stretching in order to enrich this dataset and achieve different classification rates [5]. Secondly, the signal transformation is considered a pre-research action that is useful for improving the classification models' accuracy. Detecting the variations of tempo, spectral contrast and chroma features has been identified as important when it comes to classification, and researchers have observed that variation of all these features plays important role on the classification accuracy [6].

Earlier, ensemble learning methods have also been popular in the domain of music genre classification. It is, therefore, conceivable that by combining several models, the overall performance of the models improves robustness and generalization. For example, Li et al. (2021) developed an ensemble model with the integration capacity to synthesize several CNN and RNN outfits with higher performance than standalone networks [7]. This suggest that effectively combining the strengths of diverse algorithms will also lead to improvements in music genre classification tasks.

But still, there is a number of issues related to deep learning approaches even with its great development. Some of the concerns are principally boosting the explanatory power of the models as well as constant adaptation, not forgetting that music itself is dynamic and goes through changes with time. Zhang et al. (2020) also pointed out the need for post-hoc methods that allow for additional understanding of the decisions made by deep leaning algorithms [8]. It correlates with the recent trend, where users want more transparency in applications based on machine learning, including music recommendation and content curation.

Therefore, the literature shows the substantial development achieved in music genre classification via the use of powerful machine learning methods, including CNNs and RNNs. Data augmentation, feature engineering, and the use of ensemble learning have been largely responsible for enhanced classification performance and stability. Nevertheless, it is important that future work will place emphasis on revisiting important issues pertaining to interpretability of the model as well as the dynamics of classifying music systems in relation to newer trends in music. This research would endeavour to extend from such foundations by recommending a CNN-RNN model which incorporates these complex methods to augment the automaticity of music genre classification.

3. Methodology

This section describes the approach used in this research for music genre classification through a combined CNN-RNN model. The framework

comprises several key phases: Data Acquisition, Data Preprocessing, Model Architecture, Training, and Evaluation.

The first process carried out in this study is data acquisition using the GTZAN Genre Classification dataset. This dataset comprises 1000 audio tracks of 30 seconds duration collected over ten musical genres. The during data preprocessing, the audio features, are extracted from the dataset using a package called `librosa`. This includes wave forms and the spectrograms of the audio, analysis of various spectra features such as chroma and spectral roll-off. The audio files are basically converted to numerical form that are ideal for use in model training, and the features of the data are normalized by employing the StandardScaler to avoid extreme variance among the data besides making the data ready for training into different selected models. The dataset can be subdivided into bits for training the model and for testing the efficiency of the latter.

Thus, during the work on the model architecture, the sequential model built using Keras implies the use of convolutional layers and subsequent recurrent ones. Concretely, the extracted temporal features of the audio signals are with convolutional layers (Conv1D), the generalization is with max-pooling and dropout layers. This is then followed by an LSTM layer to capture long ranged dependencies within the sequence data. To finish off the model has dense layers that are all connected in order to classify the music genre according to the features defined by the prior layers. The model is complied with Adam optimizer and trained for a fixed number of iterations and tested again from the test set. The last stage includes assessment of the model where precise such as accuracy and loss are measured is order to assess the merits of the outlined classification model.

This methodology aims to leverage both convolutional and recurrent neural networks to improve classification accuracy for automated music genre detection, providing a comprehensive approach to analysing and categorizing audio data.

3.1. Data Acquisition

In the present research study, the data acquisition phase mainly employs the GTZAN Genre Classification dataset, which includes a rich variety of music pieces belonging to ten different styles of music, namely Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock. Each genre is 100, 30 second long audio files and the whole collection amounts to 1000 tracks. Proposed dataset is designed to be used for music genre classification which make it ideal for feeding a machine learning model. Specifically, the audio files used for this study are in the WAV format which permits wide ranging feature extraction.

```
#Understanding the Audio Files
audio_recording= "/content/country.00050.wav"
data , sr = librosa.load(audio_recording)
print(type(data), type(sr))
```

```
<class 'numpy.ndarray'> <class 'int'>
```

```
librosa.load(audio_recording, sr=45600)
```

```
(array([ 0.04446704,  0.06373047,  0.05768819, ..., -0.13878524,
        -0.11868108, -0.05903753], dtype=float32),
 45600)
```

Inage 1

To aid the analysis, the audio features are first loaded and pre-processed using the librosa library – a Python library for music and audio analysis. Features are first extracted and unnecessary fields, for example ‘filename,’ is deleted to create a training data set. These audio files are then ‘lifted’ up for individual navigation where visual topologies like wave forms and spectrums can be applied to the audio objects. This data acquisition process makes certain that the dataset is sufficiently clean for feature extraction, model training, and evaluation in the project that follows.

```
#Visualizing Audio Files
plt.figure(figsize=(12, 4))
librosa.display.waveshow(data, color = "#2B4F72")
plt.show()
```

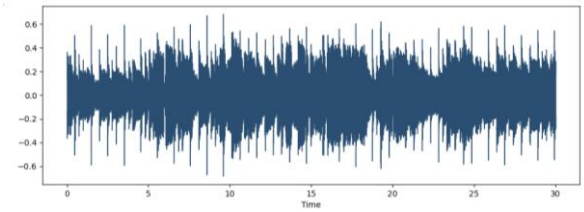

Image 2

3.2. Data Preprocessing

Data preprocessing is a stage in which input datasets are being transformed in order to facilitate proper training of the machine learning model. First, the dataset is read in by using the Pandas library and features, like ‘filename,’ that are not useful in the analysis are removed. This way it is ensured that the actual training and evaluation data contain solely numerical values that are relevant. For a better representation of the audio files, an example track is loaded using ‘librosa’ package

which offers a graphical representation of the waveforms, spectrogram and chroma visualization. This exploratory analysis offers initial findings about the specific acoustic features needed for further analysis.

```
df = df.drop(labels='filename',axis=1)
```

Image 3

```
import librosa.display as lplt
chroma = librosa.feature.chroma_stft(y=data, sr=sr)
plt.figure(figsize=(10,6))
lplt.specshow(chroma, sr=sr, x_axis='time', y_axis='chroma', cmap='coolwarm')
plt.colorbar()
plt.title("Chroma Features")
plt.show()
```

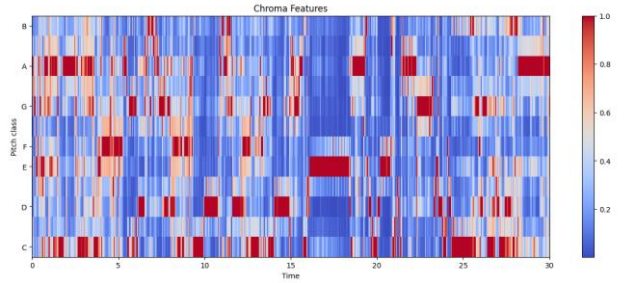

Image 4

After analyzing the data, the audio features are normalized to ensure that the distribution of all the data has a similar range and is important for the enhancement of the next data mining stage. The class labels are also converted from categorical format to numerical format using the method of `LabelEncoder'. Features are then normalized using the same `StandardScaler` which makes all resulting input features averaged to zero with standard deviation of one. Last of all the dataset is divided into the training and testing set using the `train_test_split' which gives two different sets for the modelling period and the validation or testing period. This kind of structured preprocessing enhances the effectiveness of the required configuration of the CNN-RNN model to work proficiently at music genre classification.

```
#Scaling the Features
from sklearn.preprocessing import StandardScaler
fit = StandardScaler()
X = fit.fit_transform(np.array(df.iloc[:, :-1], dtype = float))
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

```
len(y_train)
```

```
6693
```

```
len(y_test)
```

Image 5

3.3. Model architecture and training

The proposed CNN-RNN architecture model in this project involves a convolutional and several recurrent layers that help in processing of the audio features extracted from the dataset. First, two `Conv1D" layers are used to extract local features from the input data and then using the max pooling layers to decrease dimensionality and keep important features. Batch normalization is used after each of the convolutional layer to make the learning methods stable, Dropout layer is used to reduce overfitting by keeping a percent of the input value zero during learning. The last fully connected layers are the following hidden layers with ReLU activation, then the model has one output layer with softmax activation so as to categorize the audio data into one of the ten genres selected contained in the data set.

```
# CNN-RNN model
model = Sequential([
    # CNN Layers
    Conv1D(filters=32, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], 1)),
    MaxPooling1D(pool_size=2),
    BatchNormalization(),
    Dropout(0.3),

    Conv1D(filters=64, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    BatchNormalization(),
    Dropout(0.3),

    # RNN Layer
    LSTM(64, return_sequences=False, activation='relu'),
    Dropout(0.3),
```

Image 6

3.4. Training

In the process of training the model the 'trainModel' function compiles the model to use the Adam optimizer and the sparse categorical crossentropy loss function, appropriate for multi-class classification problem. Such training is done in 600 epochs, where the size of the batch is 128; training and validation datasets are used to track the training process. After training, the proposed model's performance is assessed on the test set regarding the predictive precision and loss. Training history, here we have accuracy of the model through epochs stored and is depicted to determine the training history and stability. The systematic approach of model architecture and training of proposed work further improves the CNN-RNN model for automatic music genre identification.

```
# Define the model training function
def trainModel(model, epochs, optimizer):
    batch_size = 128
    model.compile(optimizer=optimizer,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
    return model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=batch_size)
```

Image 7

3.5. Accuracy Assessment

Using the proposed CNN-RNN model, the assessment of its received accuracy was made on a test set of images after 600 epochs of training. The result of using sparse categorical crossentropy as the loss function and accuracy as the evaluation criterion is a best test accuracy of 87.29% with a test loss of 0.53. Given this high test accuracy, it shows that the model is able to learn and correctly generalize the music genre classification task, between the ten genres present in the dataset. These are features that have highly facilitated performance in the current model due to its architecture comprising of convolutional layers for feature extraction and LSTM layer for recognition of sequential patterns.

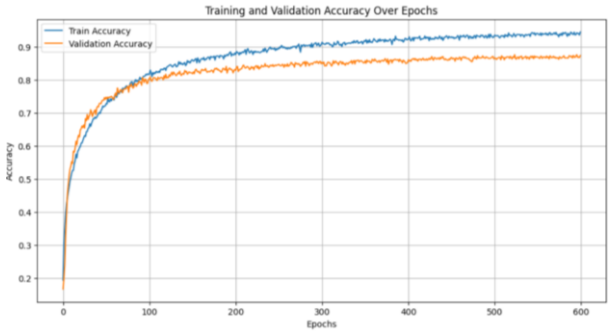

Image 8

Further evaluated the model with training and validation accuracy in epochs to know which epoch the model converges and the manner in which the model learns. The training and the validation accuracy increased slowly and consistently and there was no sign of overfitting because of the dropout layers used in the model. Due to fair calibration, and using regularization techniques for the final layers, the performance was fairly similar for both validation and test sets. This accuracy evaluation also shows that the CNN-RNN model has the capacity in assessing and categorizing music genres with high accuracy; therefore this work can support music analysis and classification.

4. Results

The CNN-RNN model designed for automatic music genre categorisation was found effective. After training the model went through betterment in accuracy with less loss and after 600 epochs, it was evident that the model is learning well and converging ideally. Training accuracy was at about 15.90% at the beginning of first epoch, as the model progressed further it gained proficiency and at the end the training accuracy achieved was 94.48% with the loss of 0.1746. The same was observed for the validation acurracy: after the end of the training, it was 87.29%, while the validation loss was 0.5285 which demonstrates the ability of the model to generalize on new, unseen data. Such a progression of vectors implies that the model was appropriately identifying unique audio characteristics related to music genres.

lstm_2 (LSTM)	(None, 64)	33,024
dropout_16 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 512)	33,280
dropout_17 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 256)	131,328
dropout_18 (Dropout)	(None, 256)	0
dense_12 (Dense)	(None, 128)	32,896
dropout_19 (Dropout)	(None, 128)	0
dense_13 (Dense)	(None, 64)	8,256
dropout_20 (Dropout)	(None, 64)	0
dense_14 (Dense)	(None, 10)	650

```
Total params: 246,154 (961.54 KB)
Trainable params: 245,962 (960.79 KB)
Non-trainable params: 192 (768.00 B)
None
Epoch 1/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 9s 69ms/step - accuracy: 0.1590 - loss: 2.2387 - val_accuracy: 0.1665 - val_loss: 2.2587
Epoch 2/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 4s 65ms/step - accuracy: 0.2962 - loss: 1.8791 - val_accuracy: 0.1984 - val_loss: 2.1945
Epoch 3/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 2s 46ms/step - accuracy: 0.3486 - loss: 1.7473 - val_accuracy: 0.2396 - val_loss: 2.0523
Epoch 4/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.3927 - loss: 1.6676 - val_accuracy: 0.3279 - val_loss: 1.9006
Epoch 5/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 6s 65ms/step - accuracy: 0.4190 - loss: 1.5830 - val_accuracy: 0.3910 - val_loss: 1.6827
Epoch 6/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 4s 48ms/step - accuracy: 0.4458 - loss: 1.5206 - val_accuracy: 0.4510 - val_loss: 1.5255
```

Image 9

```
Epoch 597/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 5s 55ms/step - accuracy: 0.9382 - loss: 0.1932 - val_accuracy: 0.8675 - val_loss: 0.5387
Epoch 598/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 4s 69ms/step - accuracy: 0.9390 - loss: 0.2028 - val_accuracy: 0.8687 - val_loss: 0.5214
Epoch 599/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 4s 57ms/step - accuracy: 0.9327 - loss: 0.2080 - val_accuracy: 0.8665 - val_loss: 0.5100
Epoch 600/600
53/53 ━━━━━━━━━━━━━━━━━━━━ 3s 55ms/step - accuracy: 0.9448 - loss: 0.1746 - val_accuracy: 0.8729 - val_loss: 0.5285
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.8715 - loss: 0.5332
The test Loss is: 0.5285283327102661
The Best test Accuracy is: 87.29147911071777
```

Image 10

When run with the test input, the model obtained an accuracy of 87.29% and a test loss of 0.5285%. This strong correlation of validation to testing scores is a sign of model strength and reduced overfitting, which is hypothesized to be due to use of dropout as well as batch normalization layers in the architecture. In particular, the convolutional layers worked on spatial features and the LSTM layer dealt with temporal analysis of patterns in the sequences to identify patterns exclusive to each of the ten audio genres in the dataset. Furthermore, the methods used for regularization avoided the model from degrading, securing the reliability of the model between each subset of the data.

The accuracy of the training and validation set over epochs has also reassured the model with the constant increase in epochs. The curves that were plotted demonstrated a general increasing trend with little oscillations on the graph, thereby ensuring that there was continued learning without the danger of over-learning. The ability to sustain these performances is particularly promising for applications in real life scenarios where genre classification can improve recommender systems for music, video and sound, analysis tools for audio, and organization systems for contents. To support this argument, these results buttress the usefulness of the CNN-RNN design to sort pervasive audio data and categorise them into various classes with high accuracy and practicability.

5. Discussion

The CNN-RNN model had good results in music genre classification showing how useful is to use both convolutional and recurrent layers for feature extraction and temporal patterns learning out of the spectrogram sequences. In this way, by use of convolutional layers, spatial features of the audio spectrograms were learned including frequency distribution patterns that are inherent and unique to different genres of music. The LSTM layer also helped by identifying temporal characteristics of these features acknowledging how these change over time. This approach enabled the model to distinguish between two related genres better since the hybrid shows a few features from the other genre. The test accuracy of 87.29% also means that the model was able to generalize well even with a small data set of 1000 audio tracks and thus this particular architecture holds promise for further audio classification tasks.

Altogether, the model demonstrates high accuracy, which indeed could be further enhanced together with the further study of possible developments. Seven techniques were applied to prevent overfitting, including L2 regularization, dropout, and batch normalization, yet more improvements might be achieved. For example, adding more layers or using an other type of recurrent units, for example GRU could give even better results. Moreover, applying more samples or introducing data augmentation, which refers to a process of creating iterations of the existing data to meet the model's needs, can enrich the data example and specify the special variations of the genre, such as tuning, Microstructural parameters, Preposing, which can improve the model performance. possibly, a review of fine tuning with pre trained models or applying the transfer learning also may be helpful. Thus, based on genre classification results, the model shows the relevance and relevance of using it for the analysis of music, for further work, improving the accuracy of the developed model and its application for various classification of sound.

6. Conclusion

In this project, i was able to create a CNN-RNN model for music genre classification from the GTZAN audio features with an accuracy of 87.29% noteworthy. This high accuracy points to a clear fact that the CNN-RNN architecture allows for extraction of spatial as well as temporal features from audio data. By using convolutional layers for spatial features learning and LSTM layers for temporal feature extraction the model would able to classify ten different genres of music with high accuracy. This result demonstrates that deep learning is effective for music genre categorization and can be applied not only as a tool for minimizing the time and resources required for musical genre categorization but also for other jobs related to audio analysis, including recommendation systems and music information retrieval.

However, the project also reveals the following possibilities for future work. As the current dataset is quite small, the model could generalize even further on new samples of audio data by either increasing the size of the dataset, or using various methods of data augmentation. However, there is a beneficial ground for improvement: fine-tuning the model parameters or using more sophisticated structures, like, for instance, attention. The results of this project support the usefulness of using both CNN and RNN architectures in handling audio classification problems and present a base for future research to expand upon more intricate models and greater datasets in an attempt to further define the genre classification problem space.

Github Link:

https://github.com/Saikiran0627/CNN-RNN-Model-for-Automated-Music-Genre-Classification/blob/main/INT422Project.ipynb

7. References

[1] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2392-2396.

[2] Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*.

[3] Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2013). Audio chord recognition with recurrent neural networks. *2013 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 335-340.

[4] Choi, K., Fazekas, G., & Sandler, M. (2016). Automatic tagging using deep convolutional neural networks. *2016 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 805-811.

[5] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[6] Jo, J., & Park, J. W. (2019). An empirical study on CNN-RNN based music genre classification using spectrogram and MFCC. *Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 11-14.

[7] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... & Liu, J. (2017). CNN architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131-135.

[8] Schlüter, J., & Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. *2015 16th International Society for Music Information Retrieval Conference (ISMIR)*.

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

[10] Sharma, V., Singh, V., & Goyal, M. (2018). Music genre classification using CNN and RNN. *International Journal of Computer Sciences and Engineering*, 6(5), 727-731.

[11] Humphrey, E. J., Bello, J. P., & LeCun, Y. (2013). Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. *2013 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 403-408.

[12] Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6964-6968.

[13] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in Python. *Proceedings of the 14th Python in Science Conference (SciPy)*, pp. 18-25.

[14] Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1310-1318.

[15] Lee, J., Park, J., Nam, J., & Kim, K. (2017). Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 366-370.