Replicating Comparisons of Shallow and Deep Neural Network Architectures for Automatic Music Genre Classification

We agree that all members have contributed to this project (both code and report) in an approximately equal manner.

Joel Grimmer
Department of Computer Science
University of Bristol
Bristol, England
ki19061@bristol.ac.uk
Joel Grimmer

Anton Wallstedt
Department of Computer Science
University of Bristol
Bristol, England
su19941@bristol.ac.uk
Anton Wallstedt

Karolina Kadzielawa
Department of Computer Science
University of Bristol
Bristol, England
zo19826@bristol.ac.uk
Karolina Kadzielawa

***Abstract*—Genre is one of the most common descriptors of music, and is used widely when communicating different styles of music. In this paper, we explore the well-known music genre classification dataset, GTZAN. We replicate the shallow and deep neural networks created by Schindler et al. in 2016, from their paper "Comparing Shallow versus Deep Neural Network Architectures for Automatic Genre Classification", after which we extend their work further by considering a more recent shallow network architecture: "Bottom-Up Broadcast Neural Network" (BBNN) created by Heo et. al in 2019. We consider shortcomings of the GTZAN dataset, often overlooked when used in MGC, in addition to criticism of Schindler et al.'s construction of the dataset and how both of these factors may affect performance. We provide results that keep in line with Schindler et al.'s discovery, displaying no significant improvement when applying a deep network architecture for classification. We further prove that using a more complex form of shallow architecture (BBNN), such that the data can remain unaugmented and thus small in size, improves the classification accuracy significantly. Our results from 5 i.i.d. trials gave classification accuracies of ∼64% for the shallow and deep networks, while the complex shallow BBNN architecture produced a ∼74% prediction accuracy.**

***Index Terms*—Music Genre Classification (MGC), GTZAN, Bottom-Up Broadcast Neural Network (BBNN), Shallow Architecture, Deep Architecture**

I. Introduction

Music genre classification (MGC) is a sub-field of music information retrieval (MIR), which relies on computational understanding of music semantics. It is a well-researched topic [1], and many approaches have been inspired by the use of Convolutional Neural Networks (CNNs) for image classification [2]. Music classification has a wide range of applications, such as genre classification, mood classification, artist identification, instrument recognition and music annotation [1]. In this paper, we will focus on genre classification, more specifically on Schindler et al.'s work on comparing how shallow networks perform against deep networks, when classifying genres on smaller datasets [3]. It is often mentioned that deeper networks are better at modelling non-linear relationships than shallow networks, however this has not been strictly established in the MIR domain [3]. Large datasets, like those commonly found in the computer vision domain, are less common in the field of MIR due to copyrighted music. Schindler et al. propose that shallow networks perform better than deep networks in MGC, when the dataset is small [3]. This line of enquiry will be studied further in this paper, in which we will attempt to first reproduce the results from [3], after which we attempt extending our work to further explore how higher prediction accuracies can be attained. We present related work in this field in Section II. We detail the well-known music classification dataset used in our study, GTZAN, and how training and validation sets were constructed in Section III, along with criticisms of the dataset as a whole. The CNN architectures and implementation details are presented in Section IV and V respectively. In Section VI we present our attempt at replicating the results from [3], followed by our training curves in Section VIII. Finally, we present the results from our extensions in Section IX, and conclude our study and suggest future work in Section X.

II. Related Work

In a study conducted by Liu et al., "Bottom-up Broadcast Neural Network For Music Genre Classification", the common approach of applying CNNs in MGC was evaluated [4]. Since the field of MGC has been inspired by the success of using CNNs in image recognition, they observed that conventional methods in MGC employ similar CNN structures used in image recognition, without modification. They postulate that this results in learning features that are not adequate for MGC. The study focuses on three benchmark datasets: GTZAN, Ballroom, and Extended Ballroom [8]–[10], and proposes a CNN architecture that takes the long contextual information into consideration, resulting in more suitable information for the decision-making layer. They observed that previous network structures of MGC mainly focus on high-level semantic features, resulting in a loss of critical low-level features. Based on their observations, they created a novel architecture called "Bottom-up Broadcast Neural Network" (BBNN), with a wide and shallow architecture. Their network has few parameters to learn, and can hence be used with a smaller dataset without any data-augmentation techniques. They achieved a classification accuracy of 93.9% on the GTZAN dataset, 93.7% on the Ballroom dataset, and finally 97.2% on the Extended Ballroom dataset.

Following the success of Liu et al. [4], among two other successful deep learning algorithms [12], [13], Rafi et al. [11] performed a comparative analysis of these studies, to aid in future work in the field of MGC. The study compares three deep learning algorithms: CNN, Recurrent Neural Network (RNN) and Convolutional Recurrent Neural Network (CRNN), namely BBNN [4], IndRNN [12] and CRNN-TF [13], respectively. They observed that each study focused on multi-scale features of audio signals, in their own ways. As noted previously, the BBNN network focused on the transmission of low-level and high-level features, using compact parameters thus meaning a reduced need for data-augmentation. IndRNN focused on learning long-term dependencies, while CRNN-TF extracted spatial dependencies on both time and frequency dimensions. Rafi et al. observed that the BBNN network had outstanding performance, however struggled to distinguish similar genres. IndRNN demonstrated no increase in accuracy, however had a remarkable efficiency in reducing training time, while keeping the trade-off in terms of accuracy to only 1%. Thus they propose that IndRNN is a promising architecture for MGC, however could be implemented with additional layers to improve classification accuracy. The CRNN-TF network lacked accuracy compared to BBNN and IndRNN, however Riu et al. proposed that replacing the long short-term memory (LSTM) layer in this network with IndRNN could improve the performance of the architecture. All networks achieved $\geq 95\%$ prediction accuracy.

A more recent study conducted by Heo et al. [5]: "Convolution Channel Separation and Frequency Sub-Bands Aggregation for Music Genre Classification", attempts to further improve conventional frameworks for MGC, by drawing inspiration from the BBNN study [4]. They used a larger dataset, the Melon Playlist [6], containing 600 times more data than GTZAN. They emphasise the importance of feature-extraction, by claiming that music has a form of hierarchical relationship between long- and short-term features. Short-term features include pitch and tempo, long-term features include melody and narrative, meaning long-term features are composed of short-term features. They propose that a model performing MGC should be able to understand this hierarchical relationship, and following from this perspective they employed ECAPA-TDNN [7] as their baseline model. The model is designed for speaker verification and has achieved state-of-the-art performance by extracting and incorporating features in various time scales. Using this model, they were able to devise a convolution channel separation technique that separates short-term features from long-term features. In addition to this, they incorporated a frequency sub-bands aggregation model, which separates the input spectrogram along different frequency bandwidths and processes each segment, based on the observation that music is composed of several instruments each residing in a different frequency band. Employing these techniques, they were able to achieve a prediction accuracy of 70.4%, a supposed improvement of 16.9% units compared to conventional frameworks.

III. Dataset: GTZAN

The dataset used in this study is GTZAN, composed by Tzanetakis [8], is one out of four datasets explored in Schindler et al. [3]. It is composed of 1000 audio files of type *.wav*, equally distributed between ten music genres: blues, classical, country, disco, hip-hop, jazz, metal, reggae and rock. The audio files are formed by 100 audio clips for each class, and each file is divided into chunks of 0.93 seconds, using a step of 50% for each chunk. This results in each chunk covering $[0.00 - 0.93, 0.47 - 1.4, 0.93 - 1.86, \dots]$ seconds, resulting in 63 chunks per audio file. From these chunks, 15 are randomly selected and converted into spectrograms, which are used as inputs to our networks. The dataset has been split up into a training set consisting of 11,750 examples (∼76%), and a validation set consisting of 3,750 examples (∼24%). The training and validation examples are created by selecting 25 random audio files and their corresponding spectrogram from each class, forming the validation set, and the remaining spectrograms are used to create the training set. Each example contains the filename, the audio spectrogram of size $[1, 80, 80]$, a randomly selected 0.93 seconds of audio from the given file, the label and finally the audio sample used to created the spectrogram.

See *Fig. 1* for some example waveforms, their corresponding spectrograms and labels. For the first set of spectrograms, the x-axis represents time, spanning from 0 to 10,000 (ms) and the y-axis represents frequency, spanning from 0.0 to 1.0 Hz. For the waveforms, the x-axis represents time, spanning from 0 to 20,000 (ms) and the y-axis represents amplitude, from -2,000 to 2,000. The authors of the reference paper reshaped these MEL spectrograms to (80x80) by down-sampling the spectrogram to a (40,80) representation and repeating the last (1,40) snapshot a further 40 times, forming (80,80).

Moreover, in a paper published in 2013, Sturm criticises the use of GTZAN for MGC [14]. He claims GTZAN was not created specifically for MGC, based on personal communications with Tzanetakis, however has been adopted into MGC research due to its availability and has thus become a benchmark dataset. As such, it is worth mentioning these issues as they may provide clarity when discussing our results in Section VIII, where we provide example cases where our models works well and less well. In [14], Sturm shows that GTZAN suffers from repetitions, clipping distortion, and mislabelled data. There are two genres that have considerably more mislabelled data than the rest: metal and rock. For example, in the case of metal, there are instances of rock music that have been incorrectly labelled. However, all genres but blues and classical have mislabelled data as well.

IV. CNN Architectures

For the sake of brevity we specify here that the Shallow, Deep, and Bottom-up Broadcast Networks all solely use convolutional layers that pad their inputs so that the output feature map has the same dimensions (not including channel count) as the input feature maps. This can be achieved by setting `padding="same"` when initialising convolutional layers in PyTorch.

A. Shallow Network

We have implemented the shallow architecture from Schindler et al. The shallow network performs two pipelines on the 80x80 input spectrogram. The left pipeline uses wide convolution and pooling kernels, 10x23 and 1x20 respectively, to learn frequency relations. The right pipeline uses tall convolution and pooling kernels of shape 21x20 and 20x1 respectively to learn time series characteristics. The resulting feature maps from each pipeline are flattened and merged. The network then features a 200 unit linear layer, and finally an output layer of 10 neurons to correspond to the 10 output classes. Dropout regularisation is applied to the 200 unit linear layer with a rate of 10%. All trainable layers, i.e. both convolutional layers and the 200 unit fully connected layer use the Leaky ReLU activation with $\alpha = 0.3$.

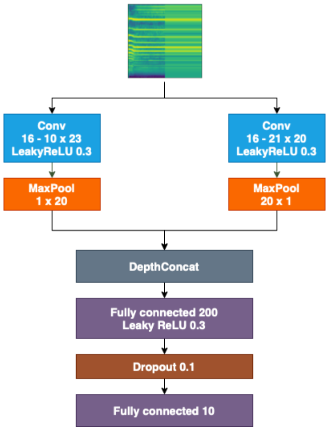

Fig. 2: Shallow CNN architecture

B. Deep Network

We have implemented the deep architecture from Schindler et al. The deep network extends the two pipelines pipelines found in the shallow network with further convolution and pooling layers. Once again the left pipeline uses wide convolution and pooling kernels to learn frequency relations and the right pipeline uses tall convolution and pooling kernels to learn time series characteristics. The resulting feature maps from each pipeline are flattened and merged. The exact kernels used in the network are shown in 3. The added pooling layers result in the concatenated feature map consisting of half the units as the shallow network. The remaining layers are identical to that of the shallow network. Once again all trainable layers use the Leaky ReLU activation with $\alpha = 0.3$.

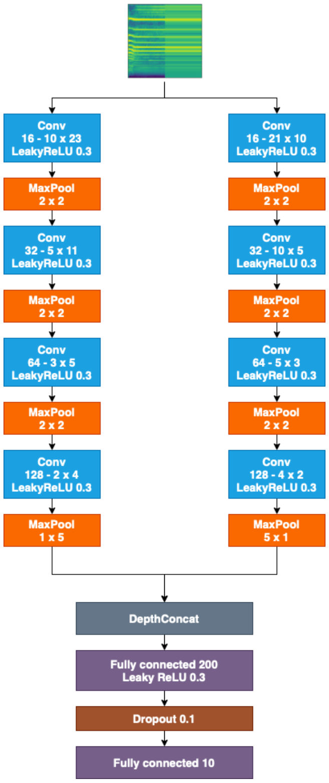

Fig. 3: Deep CNN architecture

C. Bottom-Up Broadcast Neural Network

We have implemented the Bottom-Up Broadcast Neural Network from Liu et al. [4]. The researches we combined convolutions with different kernel size to form an Inception block, a structure first devised by Szegedy et al. in Going deeper with convolutions. The researchers combined three inception blocks to learn features from multiple reception fields to create what they dubbed a ”Broadcast Module”. This module features dense connections between the blocks in order to preserve extracted feature maps to deeper layers to allow predictions to be made based on all feature-maps in the network. The network uses batch-normalisation in order to help regularise the model and reduce the need for dropout regularisation. All learnable layers are activated by ReLU, as opposed to Leaky ReLU found in the previous two models. The repeated concatenation of dense connections results in a significant number of channels in the output of the broadcast module. The researchers use Average Pooling with a stride of 2 to reduce the number of channels. Global average pooling is then used to take the average of each feature map, which the researchers say is loss prone to overfitting than traditional fully connected layers.

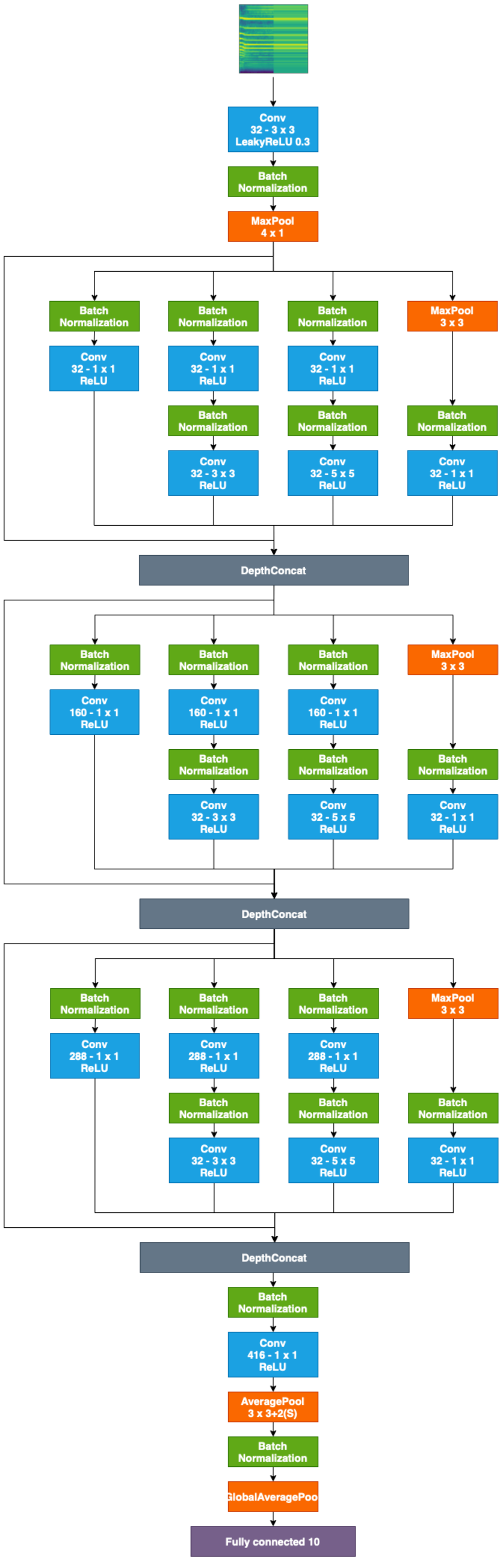

Fig. 4: Bottom-up Broadcast CNN architecture

V. Implementation Details

To replicate their results, some hyper-parameters had to be tuned since some of these were left unspecified by Schindler et al. We decided on a batch size, and did this by running 5 i.i.d. trials, sweeping over batch sizes $[16, 32, 64, 128, 265, 512]$ to see which batch size would result in the highest prediction accuracy. See Fig. 2 for our results. Our results display that there is no significance difference in using batch sizes $[16, 32, 64, 128]$, judging from a quick glance at the overlapping error bars. The worst performance was produced by using a batch size of 512, producing a prediction accuracy of $\sim$61% compared to the $\sim$65% accuracy given by the smaller sizes. To speed up the training process, we decided to use a batch size of 128, since this size is quicker than the other alternatives $16, 32, 64$, but provide the similar accuracy. We set the learning rate to $5E-5$, and used random weight initialisation, as we found that these settings in conjunction with the selected batch size produced the same results.

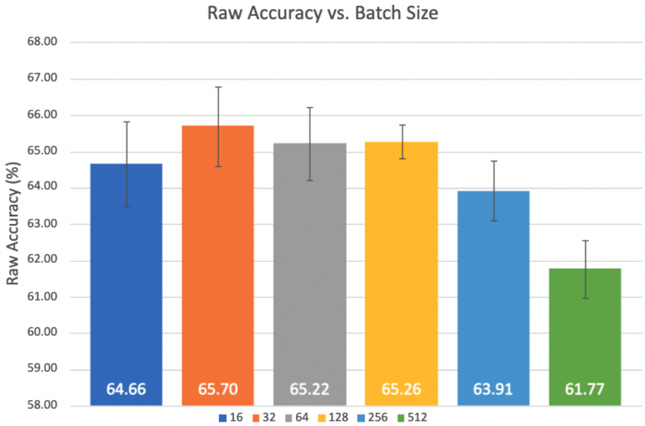

Fig. 5: Raw prediction accuracies for 5 different batch sizes: $16, 32, 64, 128, 256$ and 512. The data was retrieved from 5 i.i.d. trials. The bars display the mean prediction accuracy, with ± standard deviation.

VI. Replicating Quantitative Results

The use of random weight initialisation and stochastic optimisation algorithms, produce natural performance variations during tr4a

TABLE I: Experimental results at 100 and 200 training epochs. Mean accuracies and standard deviations of the 5-fold cross-evaluation runs calculated using raw prediction scores.

Model	***raw***	***ep***
shallow	62.96 (0.42)	100
deep	62.11 (1.02)	100
bbnn	74.37 (0.93)	100
shallow	64.19 (0.82)	200
deep	63.56 (0.55)	200
bbnn	75.41 (1.01)	200

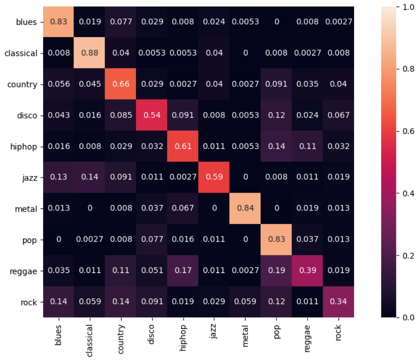

Fig. 6: Confusion matrix, Shallow Network, 200 Epochs

VII. Training Curves

VIII. Qualitative Results

Here we present example cases where our models performed well, and less well. The ground truth input and the prediction for each model is showed in Fig. 7

One of the genres the networks performed the worst on was rock, frequently misclassified as blues by ShallowNet and DeepNet and country by BBNN. Some of the confusion could be attributed to the kinship between those genres. Rock music is considered to have originated from the two genres it was most often mistaken for. Upon visual comparison between the rock spectrogram as seen in the middle of Fig. 7 and one for a blues sample Fig. 8 the similarity becomes apparent. The two share bands of high intensity around the same frequencies on the upper end for example.

What is more, the poor performance in the case of rock could at least partially be attributed to the errors in the dataset itself [14]. However the degree of the effect of that mislabeling is questionable since the predictions on metal, another frequently mislabelled genre, are more accurate than for most other genres regardless of the flaws in the dataset.

IX. Improvements

After implementing Schindler et al.'s shallow architecture, we extended our work to also include their deep network. Their deep network follows the same approach wherein two separate channels are kept for processing the frequency and time domains separately, however these channels now consist of more layers. After implementing their deep network, we extended our work even further and drew inspiration from more recent work within the MGC field, by implementing Heo et al.'s "Bottom-Up Broadcast Neural Network" (BBNN) from [4] created in 2019. This network is wide and shallow in architecture, however much bigger than Schindler et al.'s shallow implementation, and was an interesting model to enquire to further study the differences between shallow and deep architectures when classifying music. Since it attempts to make use of both short-term and long-term features, it was particularly interesting to see if this network could provide higher accuracies in comparison to Schindler et al.'s work, since the audio snippets used in training are relatively short, thus unclear whether or not the network would be able to successfully use long-term features to improve its accuracy. We hypothesised that this might lead to the network not gaining as high classification accuracy as in [4], but remained curious to see if it could improve it at all.

X. Conclusion and Future Work

Our study produced results that support Schindler et al.'s [3] hypothesis: shallow architectures are better suited for MGC when the dataset is small. Implementing their deep network produced no significant improvements on the prediction accuracy, however implementing a more complex form of shallow architecture, namely BBNN, increased our prediction accuracy significantly. From this, one can conclude that there is a benefit in using shallow architectures when applied to small datasets, avoiding the need for augmented data. However, since music evolves over time, unlike images, it is important to make use of short-term and long-term features to make the network better suited for MGC, as proposed in [4], [5].

While our extension improved performance significantly, there are additional lines of enquiry that were left unexplored in this study. One of the most important steps in deep learning is constructing the dataset, and we claim that the construction performed by Schindler et al. was sub-optimal, since they repeated the last frame of the MEL spectrogram another 40 times to produce a size of (80, 80). This repetition carries little information about the song itself, and may make it more difficult for a model to discern different genres. Moreover, shortcomings of the dataset itself should be addressed, especially issues such as mislabelled data, or repeating data. One genre our model performed the worst on was rock, and there are many instances of rock songs labelled as metal (see [14]), leading to a less distinct decision boundary between the two genres. Issues like these are likely to affect performance, and should therefore be addressed in future studies.

References

[1] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. "A Survey of Audio-Based Music Classification and Annotation" Multimedia, IEEE Transactions on, 13(2):303-319, 2011.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In "Advances in Neural Information Processing Systems" p1097-1105, 2012.

[3] Alexander Schindler, Thomas Lidy, and Andreas Rauber. "Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification". 2016.

[4] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. "Bottom-up Broadcast Neural Network for Music Genre Classification. https://doi.org/10.48550/arXiv.1901.08928. 2019.

[5] Jungwoo Heo, Hyun-seo Shin, Ju-ho Kim, Chan-yeong Lim, and Ha-Jin Yu. "Convolution Channel Separation and Frequency Sub-Bands Aggregation for Music Genre Classification". https://doi.org/10.48550/arXiv.2211.01599. 2022.

[6] Andres Ferraro, Yuntae Kim, Soohyeon Lee, Biho Kim, Namjun Jo, Semi Lim, Suyon Lim, Jungtaek Jang, Sehwan Kim, Xavier Serra, and Dmitry Bogdanov. "Melon Playlist Dataset: a public dataset for audio-based playlist generation and music tagging". https://doi.org/10.48550/arXiv.2102.00201. 2021.

[7] Brech Desplanques, Jenthe Thienpondt, and Kris Demuynck. "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification". https://doi.org/10.48550/arXiv.2005.07143. 2020.

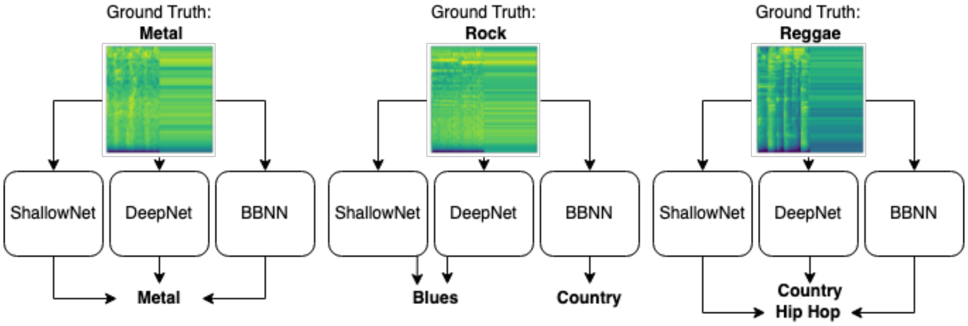

Fig. 7: The ground truth and predicted classes for metal, rock and raegge

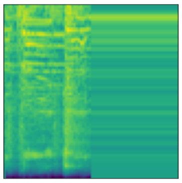

Fig. 8: Blues spectrogram

[8] George Tzanetakis, and Perry Cook. “Musical Genre Classification of Audio Signals”. In “IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp.293-302, doi: 10.1109/TSA.2002.800560. 2002.

[9] Pedro Cano, Emilia Gómez Gutiérrez, Fabien Gouyon, Herrera Boyer, Markus Koppenberger, Bee Suan Ong, Xavier Serra, Sebastian Streich, Nicolas Wack, et al. “Ismir 2004 audio description contest”. 2006

[10] Ugo Marchand and Geoffroy Peeters. “The extended ballroom dataset”. 2016

[11] Quazi Ghulam Rafi, Mohammed Noman, Sadia Zahin Prodhan, Sabrina Alam, and Dip Nandi. “Comparative Analysis of Three Imrpoved Deep Learning Architectures for Music Genre Classification”. In “International Journal of Information Technology and Computer Science” (IJITCS), vol. 13, pp.1-14. doi: 10.5815/ijitcs.2021.02.01. 2021.

[12] Wenli Wu, Guangxiao Song, Zhijie Wang, and Fang Han. “Music Genre Classification Using Independent Recurrent Neural Network”. In “Chinese Automation Congress” (CAC), pp.192-195. 2018.

[13] Zhen Wang, Suresh Muknahallipatna, Maohong Fan, Austin Okray and Chao Lan. “Music Classification using an Improved CRNN with Multi-Directional Spatial Dependencies in Both Time and Frequency Dimensions”. In “International Joint Conference on Neural Networks” (IJCNN), pp.1-8. 2019.

[14] Bob L. Sturm. “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use”. 2013.