Music Genre Classification with Convolutional Recurrent Neural Networks: An Analysis on the FMA Dataset

Chiara Maccani†, Theivan Pasupathipillai†, Carlo Sgorlon Gaiatto†

***Abstract*—The use of deep learning in Music Genre Classification has become increasingly crucial in recent years, providing powerful tools for managing and categorizing the growing music databases. In this paper, we propose several deep learning models based on Convolutional Neural Networks and Recurrent Neural Networks. We show that using a Convolutional Recurrent Neural Network results in a steeper learning curve, highlighting the efficiency of incorporating Long Short-Term Memory cells to exploit the sequential nature of music. We also experiment with a Multi-Modal architecture to investigate the effect of combining the information conveyed by different audio representations. The performance of the models are evaluated on the Free Music Archive dataset, achieving an overall accuracy of 90% and an F1-score of 60%.**

***Index Terms*—Music Genre Classification, Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory.**

I. Introduction

Music Information Retrieval (MIR) is a rapidly evolving field that deals with the analysis and organization of music data. With the increasing amount of music available online, there is a growing need for systems that can automatically categorize music based on various attributes, such as genre, style, and mood.

One of the most important tasks in MIR is Music Genre Classification, which refers to the process of automatically determining the genre of a musical piece. This task is challenging due to the subjective nature of music genres, as well as the wide variety of musical styles and influences that exist.

Recently, deep learning has emerged as a powerful approach, thanks to its capability to learn complex relationships between music audio features and genre labels, leading to improved classification performance.

Input data for deep learning in Music Genre Classification can range from hand-created features to raw waveform representation. Although the use of raw audio data requires more computational resources and a larger dataset, it takes advantage of the robust modeling capability of deep learning without being limited by hand-created features. An alternative approach is to use spectrogram-based images, as they typically retain more information than hand-created features and have lower dimensionality than raw audio data [1].

The state-of-the-art for this type of task is driven by architectures built on Convolutional Neural Networks (CNNs), which can learn to extract meaningful spatial features from the audio data, and Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM), that can capture the sequential nature of music [2].

In this paper, we introduce four models to address the challenge of Music Genre Classification and we evaluate the performance on the small subset of the Free Music Archive (FMA) dataset [3]. Firstly, we define two baseline models: a 1-D CNN that captures features from raw audio waveforms, and a 2-D CNN which takes as input spectrograms. Both models make use of a Multi-Layer Perceptron (MLP) for the final classification step. Then, we modify the 2-D baseline model replacing the MLP with a LSTM, resulting in a Convolutional Recurrent Neural Network (CRNN). Finally, we present a Multi-Modal architecture (MM-CRNN) that processes each musical piece by combining both the raw waveform and spectrogram representations, attempting to extract more informative temporal patterns.

The rest of this paper is organised as follows. In Section II, the related work and different deep learning approaches in MIR are introduced. The high-level overview of the processing pipeline is presented is Section III, while the description of the dataset and input signals is detailed in Section IV. The model architectures and the learning framework are explained in Section V, followed by Section VI in which results and performances are reported. Finally, we draw conclusions and describe potential future work in Section VII.

II. Related Work

Music Genre Classification is a widely studied task in machine learning because of its complexity and challenges. Various approaches using different types of data and deep learning architectures have been proposed over the years.

Aytar et al. (2016) [4] presented a multi-modal CNN for audio clip classification. They addressed the high dimensionality of raw audio signals by implementing a series of 1-D convolutional and max-pooling layers with large kernels and high stride values in the branch of their network that encodes the audio data. In the same year, Zhang et al. [5] proposed two ways of improving music genre classification using 2-D CNNs. They combined extensive max-pooling and average-pooling before the dense classifier to provide more statistical information to the following layers, reduce computation demands, and ensure invariance to small translations. They also implemented a residual block with a skip connection, as introduced by He et al. (2016) [6]. The use of residual blocks was shown to significantly enhance the performance of CNNs

†Department of Physics and Astronomy, University of Padova, email: {name.surname}@studenti.unipd.it

by facilitating gradient propagation, enabling the creation of deeper models capable of extracting higher-level features.

K. Choi (2017) [7] demonstrated that a hybrid architecture based on CRNNs can achieve better performance in classifying music genres on the Million Song Dataset [8] than CNNs. The authors proposed a CNN-based encoder the progressively compresses the information of the whole frequency range into one band, before feeding the resulting feature maps into a RNN. A similar analysis was conducted by Macharla et al. (2021) [9] on the GTZAN dataset [10]. They demonstrated the effectiveness of CRNNs in leveraging the spatial feature extraction capabilities of CNNs and the ability of RNNs to summarize temporal patterns.

In [11], the authors proposed multiple ensembles of CNN, RNN and CRNN to combine the advantages of different deep neural network architectures. They evaluated their approach on the small subset of the Free Music Archive, obtaining an F1-score of 54.9%.

III. Processing Pipeline

We perform several pre-processing steps on the raw audio data. First, we verify the integrity of the audio tracks, discarding any tracks that are poorly formatted or do not meet the required sampling rate and duration specifications. To cope with our limited computational resources, we randomly extract a clip of about 12 seconds from each 30-second audio track.

Then, various data augmentation techniques are applied to the raw audio data to improve the generalization performance of the models and reduce the risk of overfitting. After that, we generate a mel-spectrogram representation of the audio data and normalize both the raw audio data and the mel-spectrogram using z-score normalization. In the case of the mel-spectrogram, we apply band-wise normalization [2].

We implement two baseline models: a 1-D CNN that processes the raw audio data and a 2-D CNN that processes the mel-spectrograms. Both baseline models use a MLP for the final classification task. However, to take advantage of the sequential nature of music, we modify the 2-D CNN baseline by using LSTM instead of the MLP. Finally, we propose a MM-CRNN that combines the two baseline models by concatenating the feature maps generated by both 1-D and 2-D CNNs and feeding them into the LSTM. Several hyper-parameters are tested, such as the number of layers and neurons, dropout and weight decay.

IV. Signals and Features

The dataset we use is the Free Music Archive, a large-scale collection of music audio and annotations, specifically designed for MIR research. It provides 106,574 tracks from 16,341 artists, arranged in a hierarchical structure of 161 genres, to which a large amount of metadata is associated. Given that our computational resources are low, we limit our analysis to the use of the small version of the dataset. It is a balanced subset of 8,000 tracks distributed in eight different root genres: Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop and Rock. Since the labeling procedure was carried out by the artists themselves, who could not be necessarily objective, the presence of some noise has to be kept in mind. The tracks consist of clips of about 30 seconds encoded in mp3 format, most of them with sampling rate of $44,100$ Hz.

As mentioned in Section III, the dataset is analyzed to remove corrupted audio files and those that have less than 524288 (2^{19}) sample points, which is approximately equivalent to 12 seconds. After the cleaning procedure, the effective size of the dataset is 7994. Fig. 1 shows the waveform representation of an audio clip in our dataset.

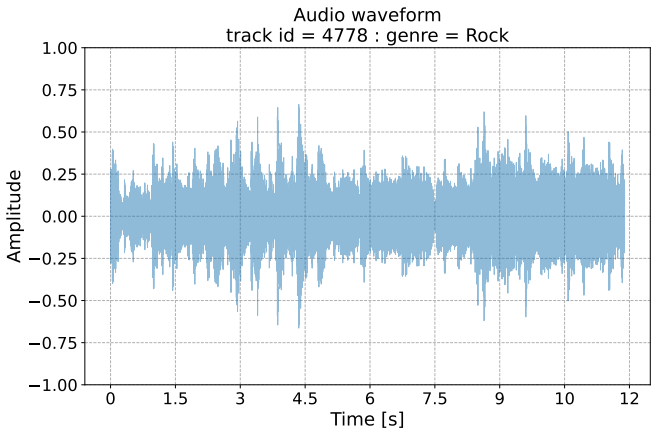

Fig. 1: Waveform representation of an audio clip labelled as Rock.

The proposed 2-D models operate on spectrograms, which show the frequency spectrum of a signal over time. They are made using the Short-Time Fourier Transform (STFT), which involves dividing the signal into overlapping segments and applying the Fast Fourier Transform (FFT) to each of them. The segments are created with a window of 4096 samples that slides with a hop of 1024 samples and uses Hann's window function to reduce the amplitude of discontinuities at the boundaries. To more accurately reflect how humans hear and perceive sound, it is useful to use non-linear frequency scales in audio analysis. In particular, the mel scale is designed so that pitch intervals are perceived as equally spaced by the human auditory system. Therefore, we create mel-spectrograms by applying a bank of 128 overlapping triangular filters that calculate the energy of the spectrum in each band [1]. The shape of each mel-spectrogram is $[128, 512]$. Furthermore, the power amplitudes of the mel-spectrograms are converted to the decibel (dB) scale, which compresses the dynamic range of the signal, making it easier to see the relative strengths of the different frequency components in the spectrogram. Fig. 2 shows the corresponding mel-spectrogram representation of the audio clip displayed in Fig. 1.

Finally, the dataset is randomly split into training (80%), validation (10%), and test (10%) sets.

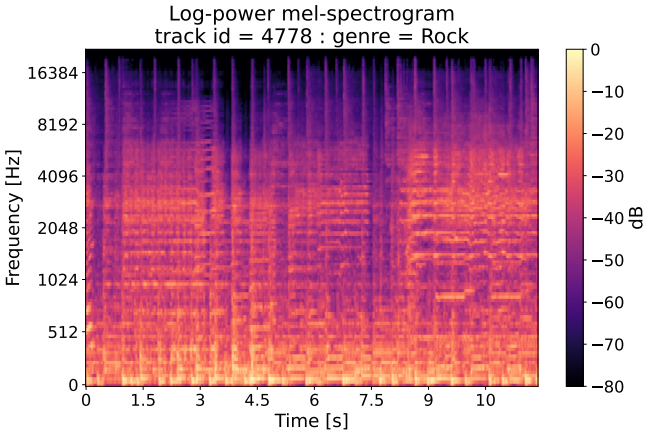

Fig. 2: Log-power mel-spectrogram of an audio clip labelled as Rock.

V. Learning Framework

We utilize the `librosa` library [12] to load audio tracks and extract random clips of 524288 samples (2^{19}) during each training iteration. This strategy serves two purposes. Firstly, using the entire signal length as input can significantly increase computational costs and impact the design of the architectures, particularly for 1-D models. Secondly, the random subsampling acts as a form of time-shift data augmentation, allowing the models to recognize the genre of a musical piece regardless of the exact starting point of the clip.

We apply other data augmentation transforms to the raw waveforms in order to increase the size of our training set and to introduce additional diversity, which can be beneficial in terms of generalization performance and to reduce overfitting. Firstly, we introduce small perturbations in the raw signal by adding gaussian noise, which amplitude is sampled uniformly from the range $[0.001; 0.015]$. Furthermore, to address the fact that musical pieces within the same genre can have a wide range of BPM, we apply a time stretch transformation, i.e. the speed of the audio signal is changed without modifying the pitch. The rate of change is sampled uniformly from the range $[0.8, 1.25]$. A rate below 1 corresponds to a slowing down of the audio while a rate greater than 1 corresponds to a speed up. The total length of the transformed sample is kept to be the same as the input one. All the transformations are carried out using the `audiomentation` library [13] and applied with probability $p = 0.5$.

Before being processed by the 1-D models, the raw audio data are normalized by applying the z-score method. It consists in subtracting to each sample x the mean and dividing it by the standard deviation, both evaluated on the whole training set:

$$x' = \frac{x - \mu_{train}}{\sigma_{train}}$$

Regarding mel-spectrograms, the z-score normalization is applied to each mel-band separately because value distributions differ significantly between frequency bands.

The architectures we propose to perform genre recognition on the pre-processed data are the following:

A. Baseline 1-D: Convolutional Neural Network

We implement a baseline model that processes the raw audio data. It has three components: an encoder, a bottleneck, and a classifier, as can been seen in Fig. 3.

The encoder is a CNN consisting of four stages, each composed of a convolutional layer, a batch normalization layer, and a max-pooling layer. The first stage uses large kernel size and stride to reduce the high sample rate input waveform, while the subsequent stages gradually increase the number of channels and decrease the feature map sizes.

The bottleneck performs global pooling on the entire temporal axis. It summarizes temporal information that would otherwise not be retained by the fully connected classifier. Specifically, it consists of max- and average-pooling operations concatenated together to provide more statistical information to the following layers, as suggested in [5].

The classifier is a MLP made up of two hidden layers with dropout, and an 8-neuron output layer to classify the audio into different genres. We use ReLU as the activation function in all convolutional stages and fully connected layers except for the last layer where we apply the softmax function.

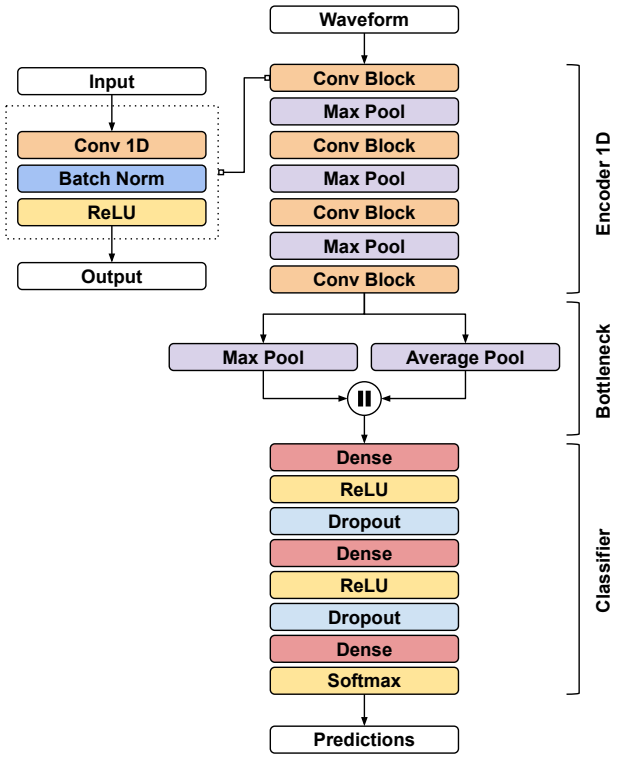

Fig. 3: Architecture diagram of the 1-D Convolutional Neural Network.

B. Baseline 2-D: Convolutional Neural Network

We implement a baseline model that processes mel-spectrogram. It has the same bottleneck and classifier as the 1-D baseline, but it uses a different encoder to handle 2-D input data of lower dimensionality, as can been seen in Fig. 4. In particular, it consists of 4 residual blocks that extract an increasing number of feature maps, interspersed with max pooling layers to perform subsampling and enhance translation invariance. Each residual block consists of 2 convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. We then skip these two convolutional operations and add the input directly before the final ReLU activation function [6]. The skip connection is implemented using a convolutional block to transform the input into the desired shape for the addition operation. The encoder is designed such that it compresses the whole frequency range into one band.

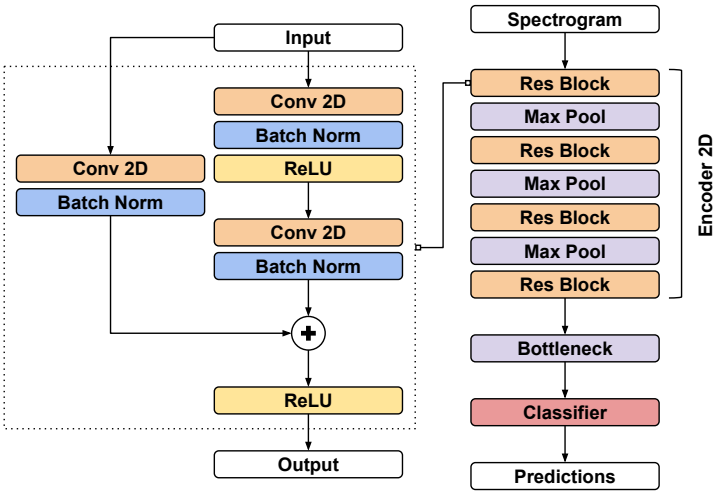

Fig. 4: Architecture diagram of the 2-D Convolutional Neural Network.

C. Convolutional Recurrent Neural Network

To exploit the temporal information that characterizes music data, we modify the 2-D baseline by replacing the bottleneck and classifier with a single-layer LSTM, shown in Fig. 5. The LSTM receives as input a sequence of 128-D feature vectors, where each dimension corresponds to an encoder feature map, and has a hidden size of 128. It inherently captures the temporal dependence of the inputs and allows an unlimited receptive field, which could be beneficial in terms of performance. Furthermore, the LSTM uses gates and memory cells to regulate the flow of information, effectively mitigating the gradient problems that affect vanilla RNNs. We then use a dropout layer to reduce the risk of overfitting before applying a fully connected 8-neuron layer with a softmax function to perform classification.

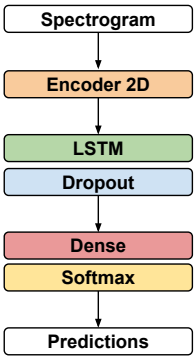

Fig. 5: Architecture diagram of the Convolutional Recurrent Neural Network.

D. Multi-Modal Convolutional Recurrent Neural Network

Combining features from raw audio and mel-spectrogram can be advantageous as they encode different types of information. We propose a multi-modal architecture that processes both data representations in parallel, depicted in Fig. 6, aiming to improve classification performance.

We use the 1-D CNN baseline to extract features from raw audio and the 2-D CNN baseline to process the mel-spectrogram. The resulting features are then merged by concatenating them along the channel dimension and fed into a LSTM, similar to the one described previously.

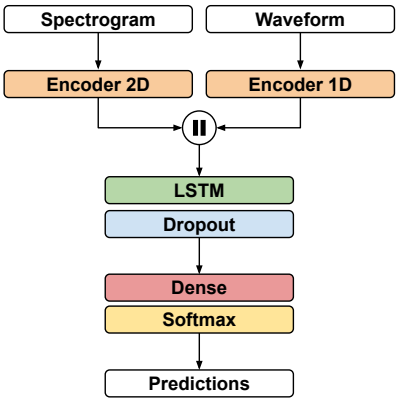

Fig. 6: Architecture diagram of the Multi-Modal Convolutional Recurrent Neural Network.

For all models we use multi-class cross-entropy loss and Adam [14] optimizer with a learning rate of 10^{-3} and specific weight decay for each architecture. We use a batch size of 32 and training is carried on until an early stopping condition on validation loss is met.

Note that during the design of the architectures, we tried different configurations by changing the number of layers and neurons in order to find a trade-off between the complexity of the model and the risk of overfitting. To further minimize the risk of overfitting and improve the generalization capabilities of the models on unseen data, we performed a grid search on the weight decay parameter and dropout rate. Weight decay regularization helps prevent overfitting by adding a penalty

term to the loss function, forcing the model to have smaller weights, while dropout is used to randomly remove a portion of the neurons during training, allowing the model to learn multiple independent representations.

For more details on the selected hyper-parameters of the architectures, see the GitHub repository [15].

VI. Results

We use several metrics to get a comprehensive understanding of the performance of our models. In particular, we calculate the accuracy, precision, recall, and F1-score on the test set. They are defined as follows:

$$\text{accuracy} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

$$\text{precision} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FP_i}$$

$$\text{recall} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FN_i}$$

$$\text{F1-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where TP_i, TN_i, FP_i and FN_i refer to True positive, True negative, False positive and False negative, respectively. The variable k indicates the number of classes. It is important to note that since our dataset is balanced, we utilize macro averaging. This approach calculates the metric for each class and then takes the average, giving equal weight to each class.

In Fig. 7 and Fig. 8 we present the loss and F1-score as a function of the training epochs, while in Tab. 1 we show the performance of the models evaluated on the test set.

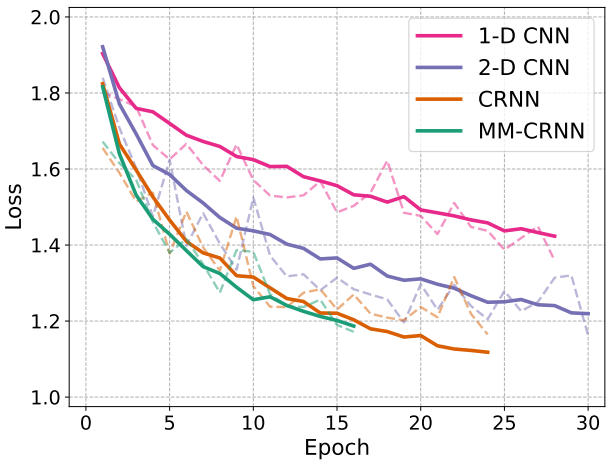

Fig. 7: Loss as a function of the training epochs. Lines with the same color belong to the same model. Solid lines represent performance on training set, while dashed lines correspond to performance on validation set.

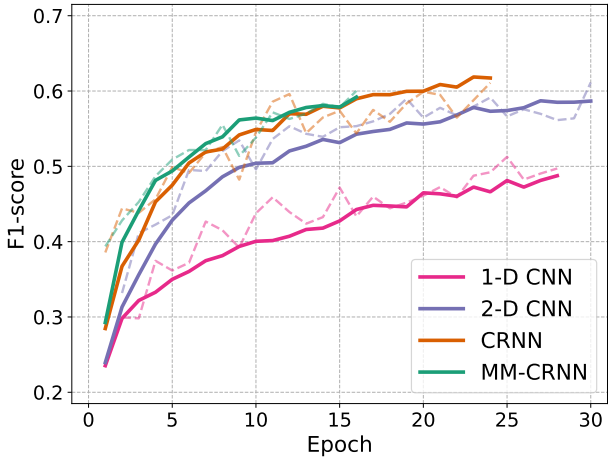

Fig. 8: F1-score as a function of the training epochs. The interpretation of the different colors and line styles is the same as in Fig. 7.

model	accuracy	precision	recall	F1-score
1-D CNN	0.88	0.54	0.54	0.54
2-D CNN	0.90	0.59	0.60	0.60
CRNN	0.90	0.60	0.61	0.60
MM-CRCNN	0.89	0.58	0.57	0.58

Tab. 1: Performance of the models evaluated on the test set.

We observe that the 1-D CNN is the worst performing model compared to the other models, which attain almost similar results. The CRNN stands out by achieving the same performance as the 2-D CNN, but in fewer epochs, highlighting the effectiveness of incorporating the LSTM module in capturing the sequential nature of musical audio tracks. On the other hand, MM-CRNN performs similarly to CRNN but it has a significantly higher number of parameters, which leads us to conclude that the use of a multi-modal approach is not justified in our case.

It is worth noting that for all our models there is a considerable difference between accuracy and other metrics. As can be seen from the CRNN confusion matrix in Fig. 9, high accuracy is the result of a large number of true negatives, while low precision and recall are related to a large number of false positives and false negatives, respectively. In the case of music genre classification, the requirements and consequences of false positive or false negative predictions depend on the specific use case. If the real-world task is to simply categorize music tracks into different genres for the purpose of organizing and filtering music collections, then a high accuracy might be sufficient and precision and recall might not be critical factors. However, if the task involves recommending music to users based on their preferred genres, or if the genre labels are used to inform decisions in the music industry, then a high accuracy might not be enough, and it would be important to have high precision and recall.

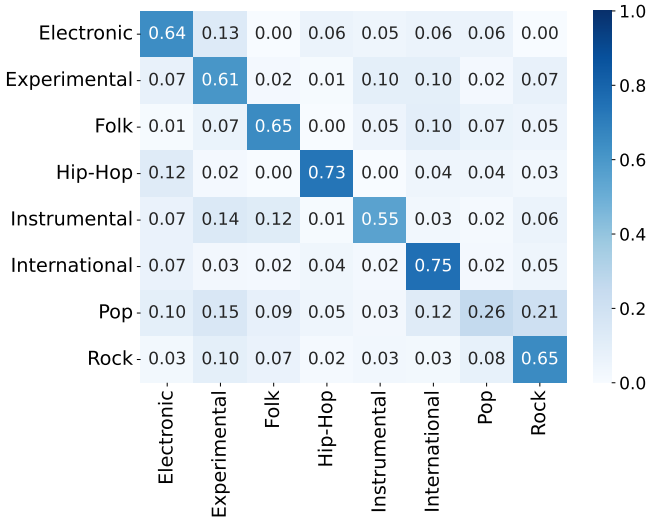

Fig. 9: Confusion matrix of the CRNN evaluated on the test set.

Note that in unbalanced datasets, accuracy might be misleading as a performance metric, since highly populated classes will have higher weight compared to the smallest ones. However, this is not our case, as describe in Section IV. Rather, we have high accuracy for most genres but low precision and recall for few of them, like Pop and Instrumental. Therefore, in future work it may be useful take steps to improve precision and recall for lower-performing classes, such as by implementing class-specific data augmentation techniques.

VII. Concluding Remarks

In conclusion, the study presented in this paper investigated the use of four deep learning models for Music Genre Classification by testing them on the small subset of the Free Music Archive dataset. The models included a 1-D Convolutional Neural Network, a 2-D Convolutional Neural Network, a Convolutional Recurrent Neural Network, and a Multi-Modal Convolutional Recurrent Neural Network.

The results showed that models incorporating Long Short-Term Memory cells performed similarly to those using only Convolutional Neural Networks, but in fewer training epochs. This highlights the potential benefits of using recurrent-based methods in music genre classification tasks and confirms the validity of this approach for future work in this area.

Further research could explore the use of these models on larger datasets, having more computational resources available, or attempt a patch-wise approach applied to audio tracks to increase the total number of clips seen by the models at each epoch, allowing the use of even deeper architectures without being limited by the overfitting problem we faced in our research.

References

[1] D. Ćirić, Z. Perić, J. Nikolić, and N. Vučić, "Audio Signal Mapping into Spectrogram-Based Images for Deep Learning Applications," in *20th International Symposium INFOTEH-JAHORINA (INFOTEH)*, (San Francisco, CA, USA), pp. 1–6, Mar. 2021.

[2] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, "Deep learning for audio signal processing," *Journal of Selected Topics of Signal Processing*, vol. 13, pp. 206–219, May 2019.

[3] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, (Suzhou, China), Sept. 2017.

[4] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning Sound Representations from Unlabeled Video," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, (Barcelona, Spain), pp. 892–900, Dec. 2016.

[5] W. Zhang, W. Lei, X. Xu, and X. Xing, "Improved Music Genre Classification with Convolutional Neural Networks," in *Proc. Interspeech 2016*, (San Francisco, CA, USA), pp. 3304–3308, Sept. 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), pp. 770–778, June 2016.

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

[8] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pp. 591–596, Jan. 2011.

[9] V. Macharla and P. Radha Krishna, "Music Genre Classification using Neural Networks with Data Augmentation A Make in India Creation," *Journal of Innovation Sciences and Sustainable Technologies*, vol. 1, pp. 21–37, Jan. 2021.

[10] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, pp. 147–172, Apr. 2014.

[11] D. Kostrzewa, P. Kaminski, and R. Brzeski, "Music genre classification: Looking for the perfect network," in *21st International Conference Computational Science – ICCS*, (Krakow, Poland), pp. 55–67, June 2021.

[12] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Python in Science Conference*, (Austin, TX, USA), pp. 18–24, Jan. 2015.

[13] I. Jordal, "audiomentations." https://github.com/iver56/audiomentations, 2019.

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, Dec. 2014.

[15] "Project code." https://github.com/carlosgorlongaiatto/NNDL˙Project, 2023.