Neural Networks and Deep Learning

Project

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

Authors

Chiara Maccani - Theivan Pasupathipillai - Carlo Sgorlon Gaiatto

Introduction

Music Genre Classification is an important task in Music Information Retrieval.

Deep learning has emerged as a powerful approach, thanks to its capability to learn complex relationships between music audio features and genre label.

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
- Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM), that can capture the sequential nature of music.

Dataset

Free Music Archive

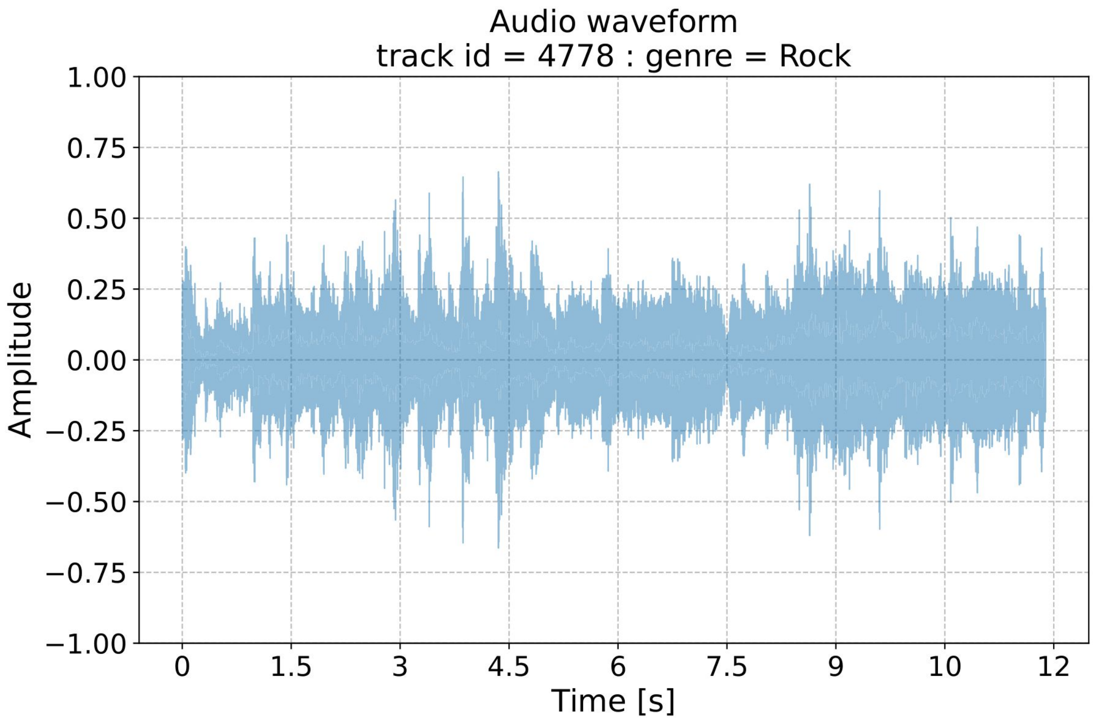
- Full dataset: unbalanced 106,574 tracks, arranged in a hierarchical structure of 161 genres.
- Small dataset: balanced 8,000 tracks, 8 root genres (Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop, Rock).

The tracks consist of clips of about 30 seconds encoded in mp3 format, most of them with sampling rate of 44,100 Hz.

Note: the labeling procedure was carried out by the artists themselves, so the presence of some noise has to be kept in mind.

Dataset: Waveform

Waveforms show the amplitude of a audio signal over time:

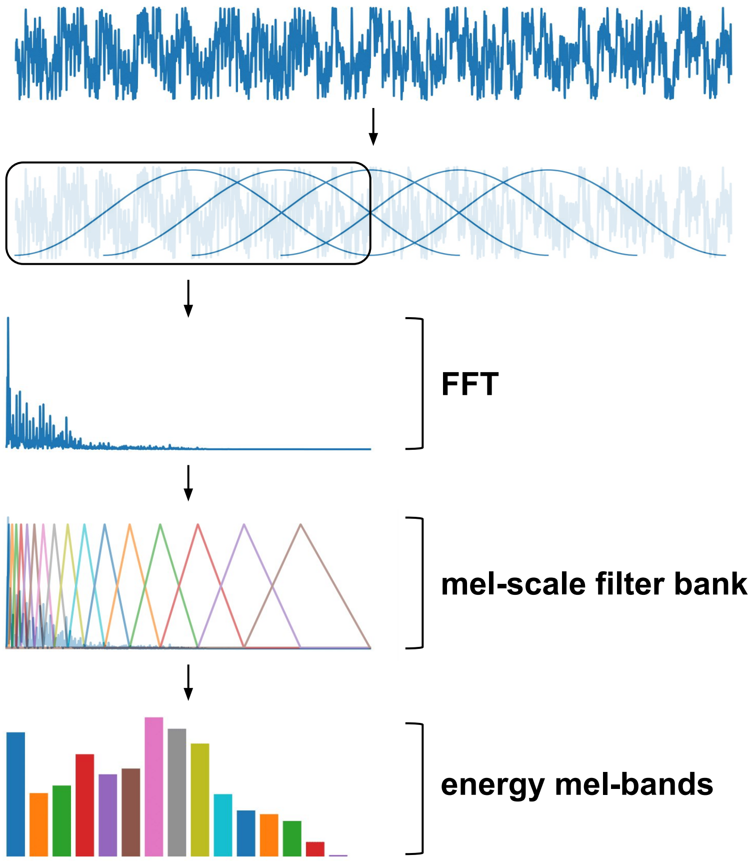


Dataset: Spectrogram

We create spectrograms using the **Short-Time Fourier Transform** (STFT) with a window of 4096 that slides with a hop of 1024 and uses Hann's function.

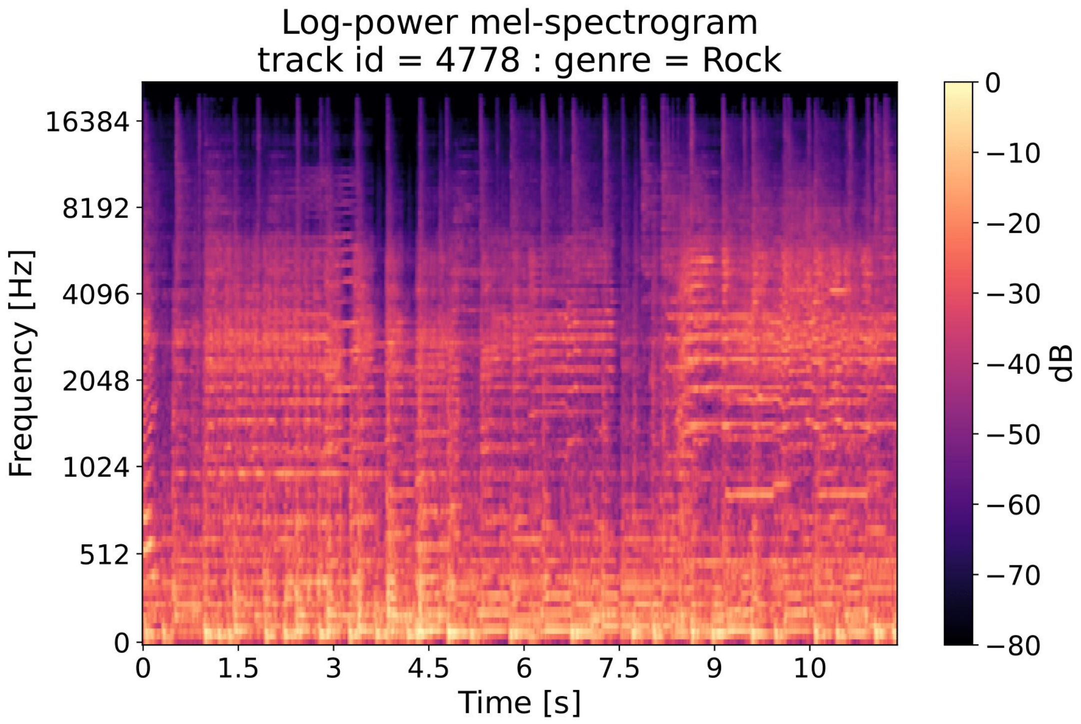
To reflect how humans perceive and hear sound, we represent frequencies in the **mel-scale** using 128 mel-bands (lin-power mel-spectrograms).

Finally, power amplitudes are converted to the **decibel-scale** (dB) to compress the dynamic range of the signal (log-power mel-spectrograms).



Dataset: Spectrogram

Spectrograms show frequency spectrum of a signal over time.



Method

The **pre-processing pipeline** consists of several steps:

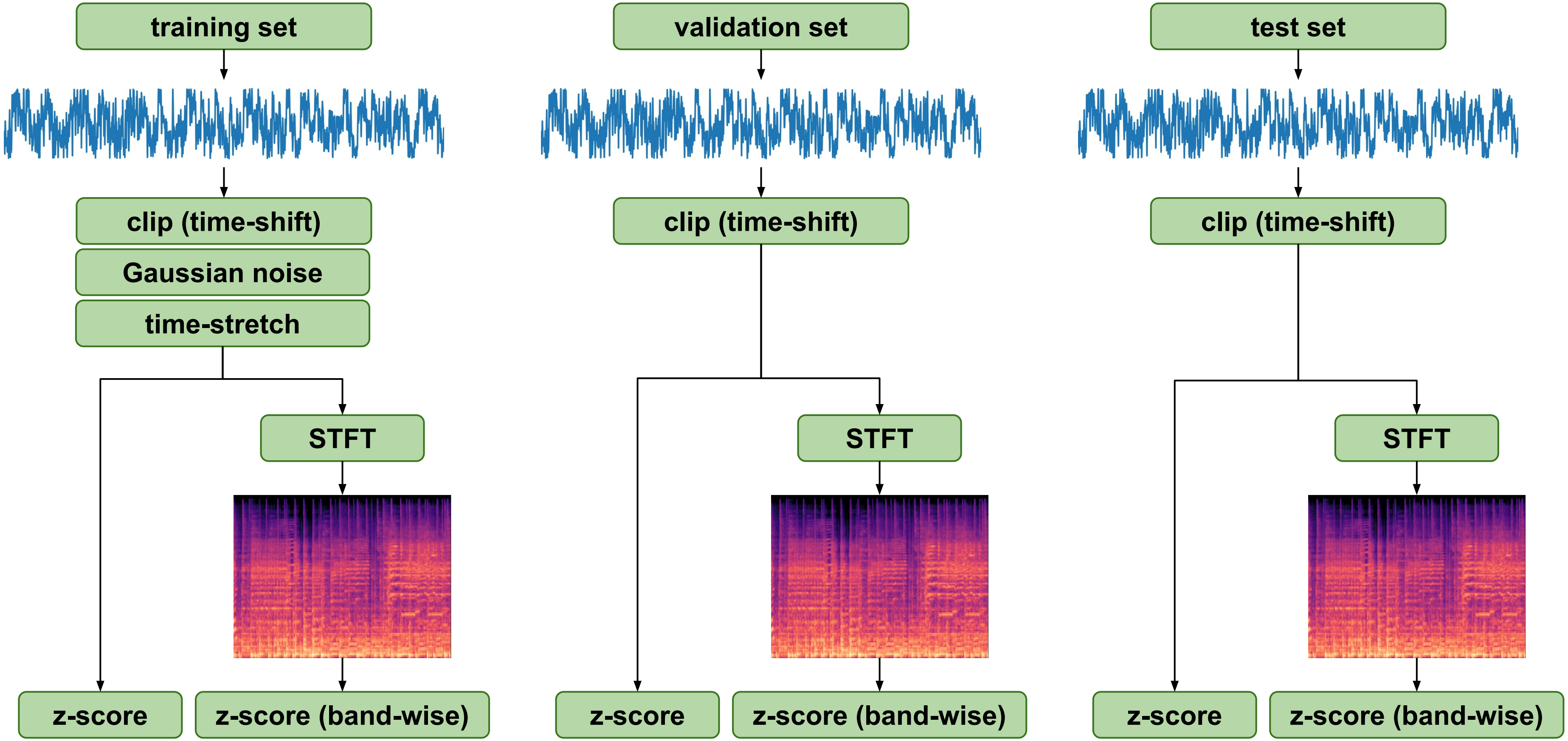
- Data cleaning: we discard tracks that are poorly formatted or do not meet the required duration specifications.

- Data splitting: we split the dataset into training (80%), validation (10%) and test sets (10%).

- Data augmentation: we extract random clips of 2^{19} samples and apply transformations to augment the data, leveraging the properties of music tracks.

- Data normalization: we apply z-score normalization using mean and standard deviation computed across the entire training set. In the case of the mel-spectrogram, we apply band-wise normalization.

Method: Schema



Models

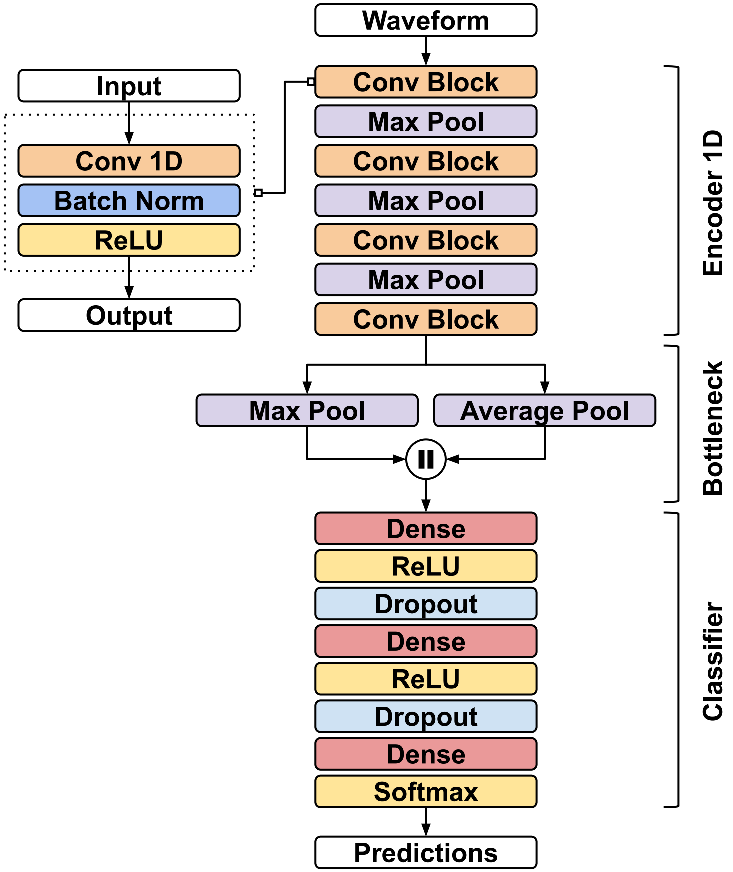
We propose 4 different models to perform Music Genre Classification on the pre-processed data:

- **Baseline 1-D: Convolutional Neural Network** (1-D CNN)
- **Baseline 2-D: Convolutional Neural Network** (2-D CNN)
- **Convolutional Recurrent Neural Network** (CRNN)
- **Multi-Modal Convolutional Recurrent Neural Network** (MM-CRNN)

Models: Baseline 1-D

The **1-D baseline model** processes raw audio data and it has three components:

- The encoder is a 1-D CNN consisting of four stages used to reduce the dimensionality and extract features.
- The bottleneck performs global pooling on the entire time axis to summarize temporal information.
- The classifier is a Multi-Layer Perceptron (MLP) with softmax function as the last activation.

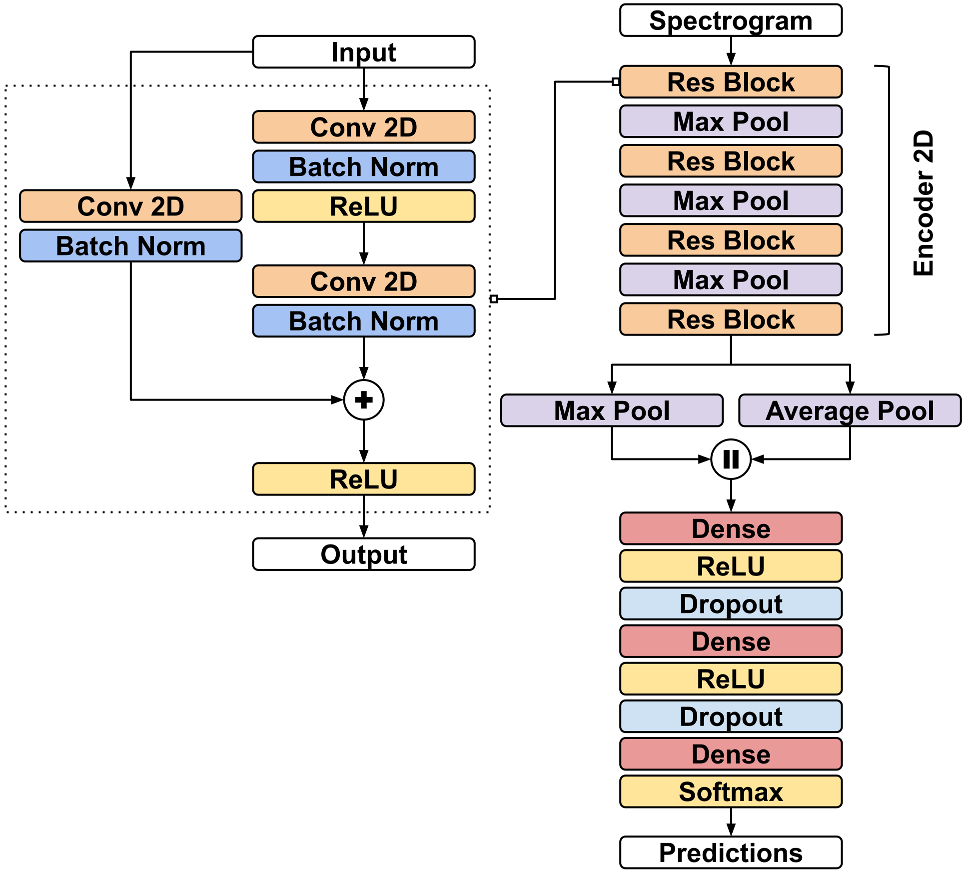


Models: Baseline 2-D

The **2-D baseline model** is made of the same bottleneck and classifier as the 1-D baseline model, but uses a different encoder to process the mel-spectrogram.

Specifically, the <u>encoder</u> is a 2-D CNN consisting of 4 residual blocks used to extract an increasing number of feature maps.

The encoder is designed to compress the entire frequency range into a single band.



Models: Recurrent Neural Network

A **Recurrent Neural Network** (RNN) is able to keep memory of the past inputs, encoding temporal features in its internal state.

A many-to-one architecture can be exploited to perform music genre classification.

We use Long Short-Term Memory (LSTM) to address the vanishing gradient problem and learn long term dependencies.

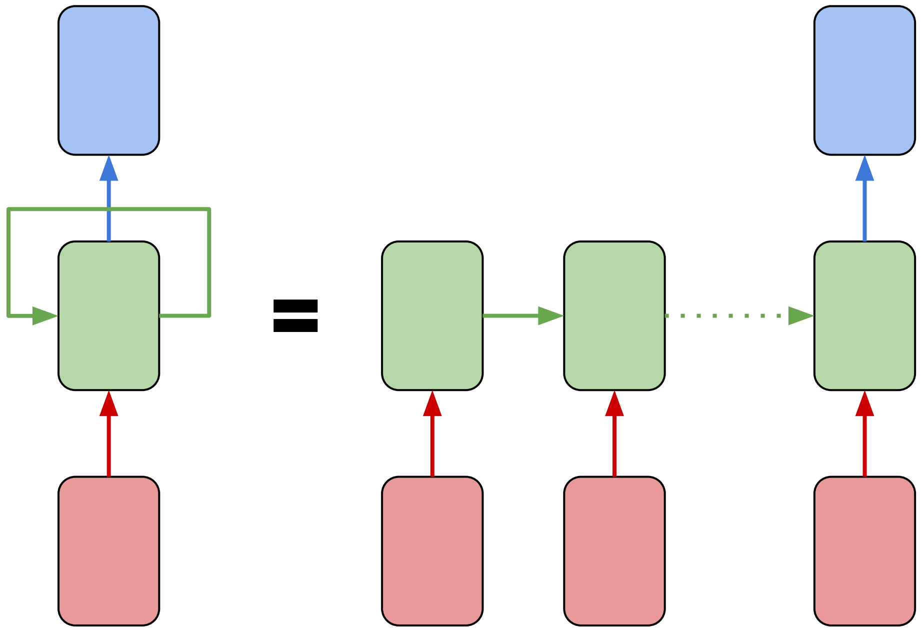
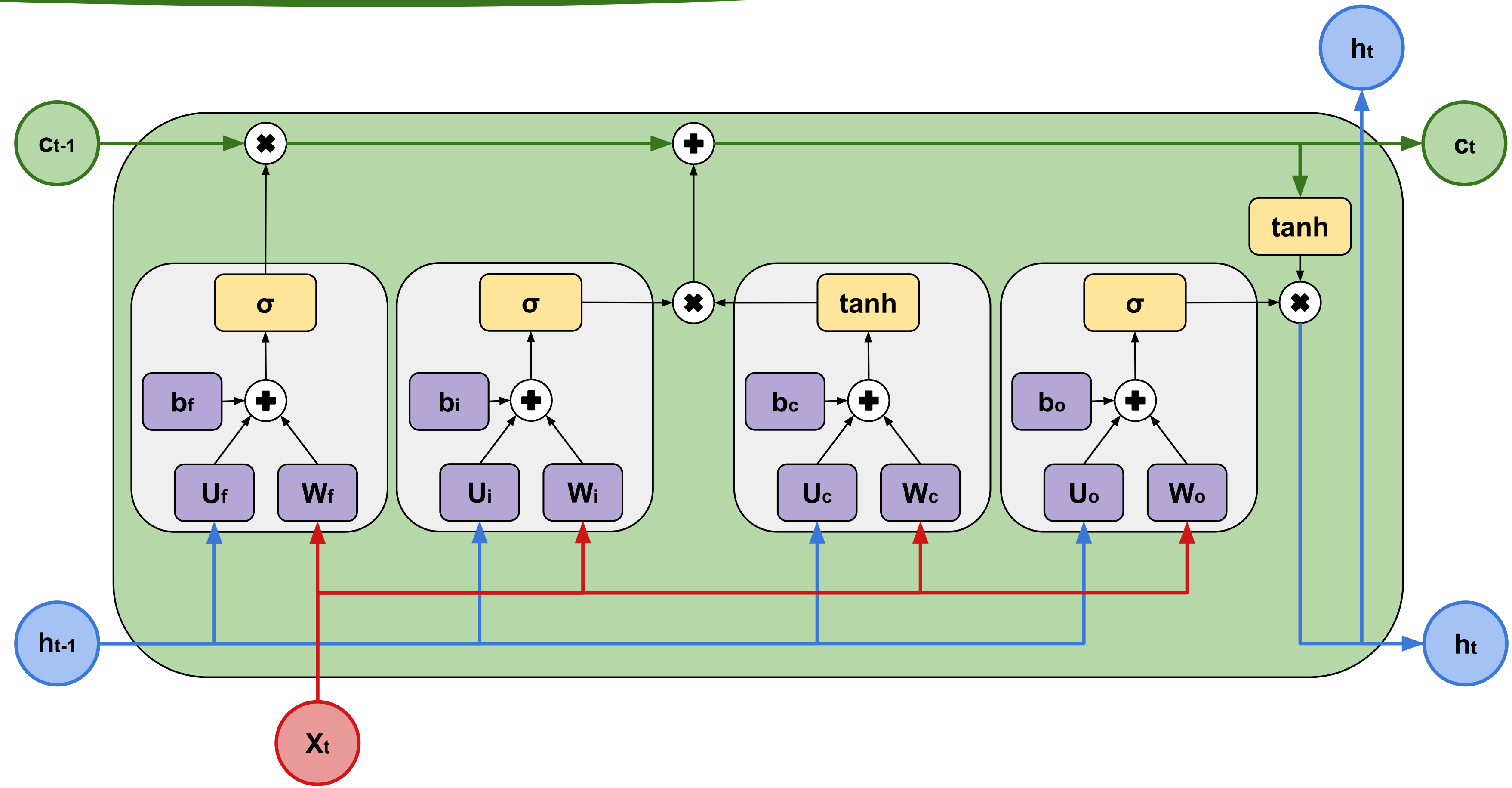


Image inspired by Andrej Karpathy

Models: LSTM

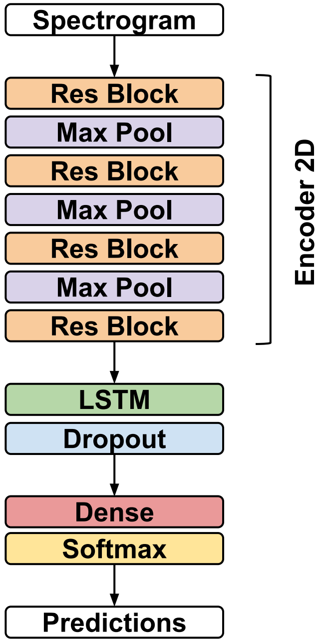


Models: CRNN

The **Convolutional Recurrent Neural Network** (CRNN) is built starting from the 2-D baseline model by replacing the bottleneck and the classifier with a single-layer LSTM.

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.

A dropout layer is added to reduce the risk of overfitting before applying a fully connected 8-neuron layer with a softmax function to perform classification.

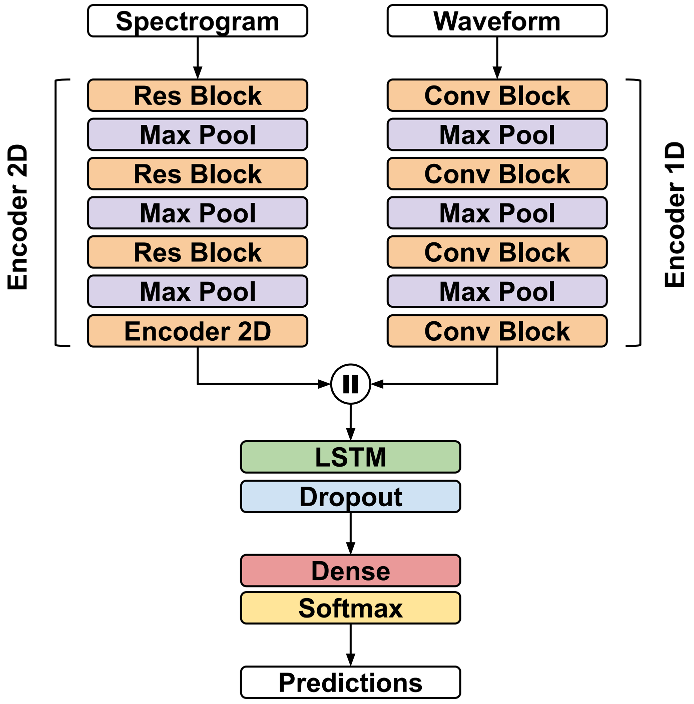


Models: MM-CRNN

We finally propose a **Multi-Modal Convolutional Recurrent Neural Network** (MM-CRNN) that processes both data representations in parallel.

We use the 1-D encoder to extract features from raw waveform and the 2-D encoder to process the mel-spectrogram.

The resulting feature maps are then merged by concatenating them along the channel dimension and fed into a LSTM.



Training

For all models we use **multi-class cross-entropy** loss and **Adam** optimizer with initial learning rate of 10^{-3} and weight decay optimized for each model.

We evaluate the model with this set of metrics:

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

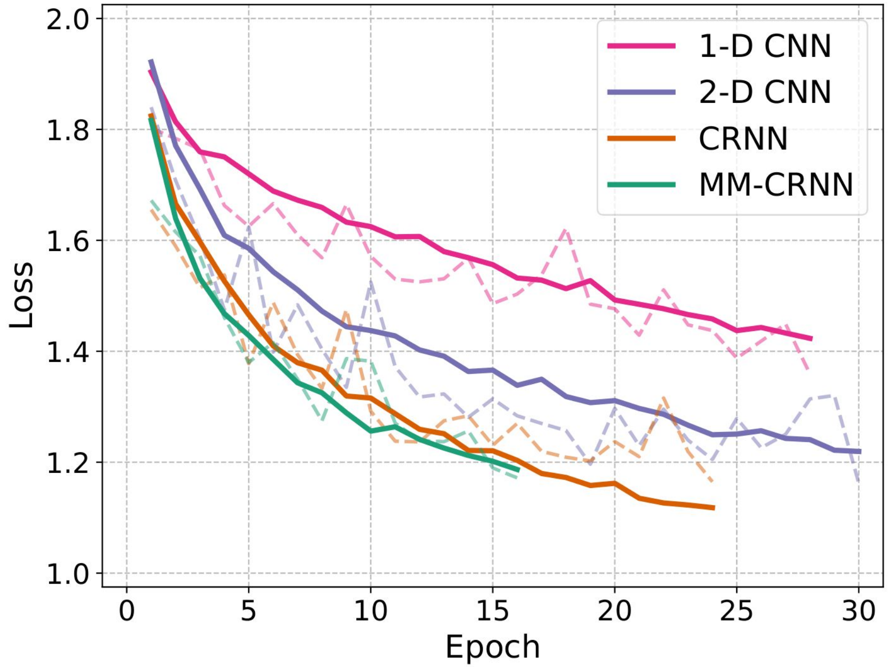
$$\text{precision} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FP_i}$$

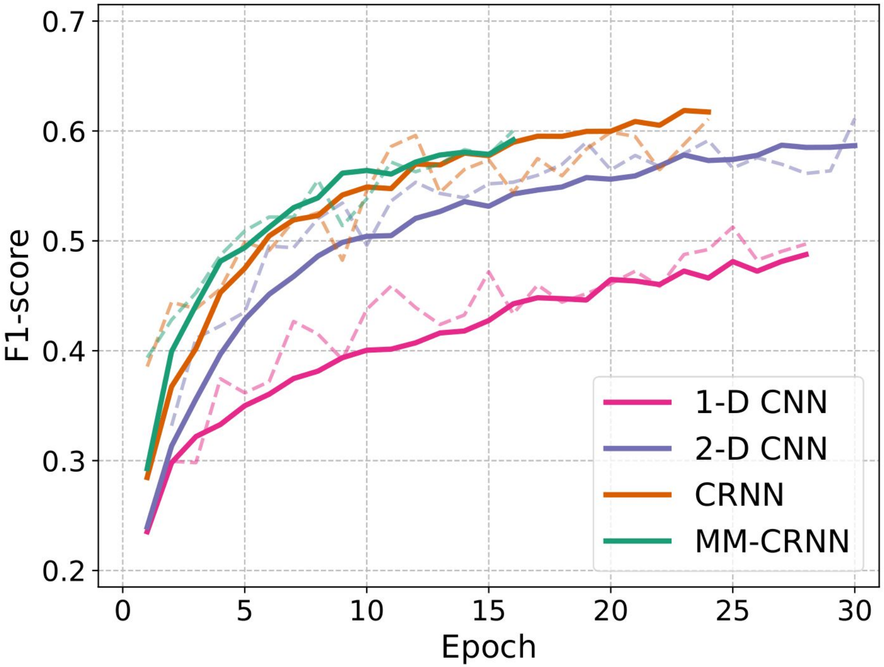
$$\text{recall} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FN_i}$$

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

Training: Learning curves

Loss and **F1-score** as a function of the training epochs:





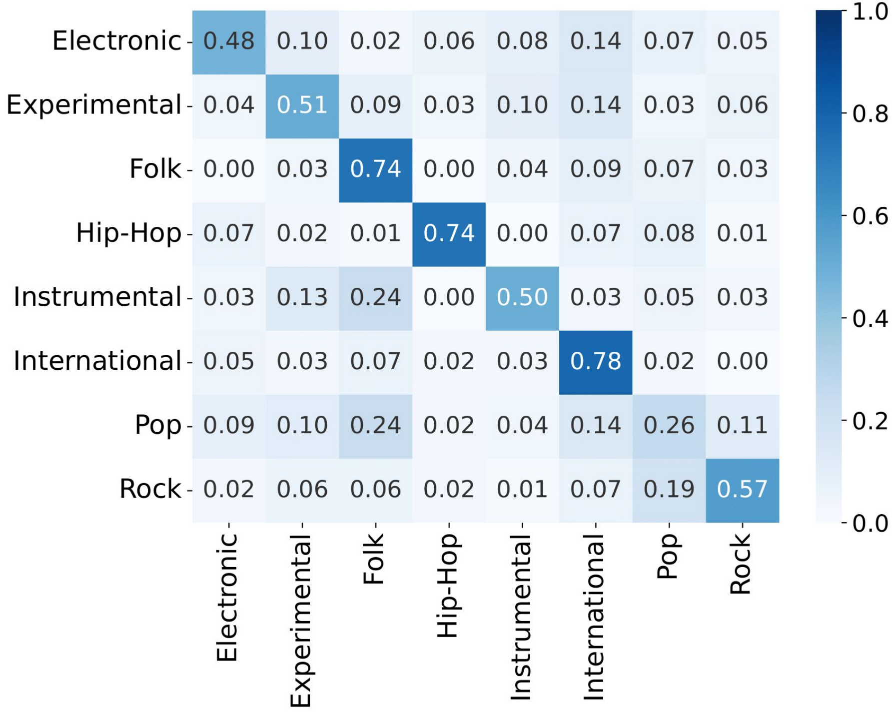
Results: Metrics

Here are the results obtained in the test set:

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-CRNN	0.89	0.58	0.57	0.58

Results: Confusion matrix

RCNN



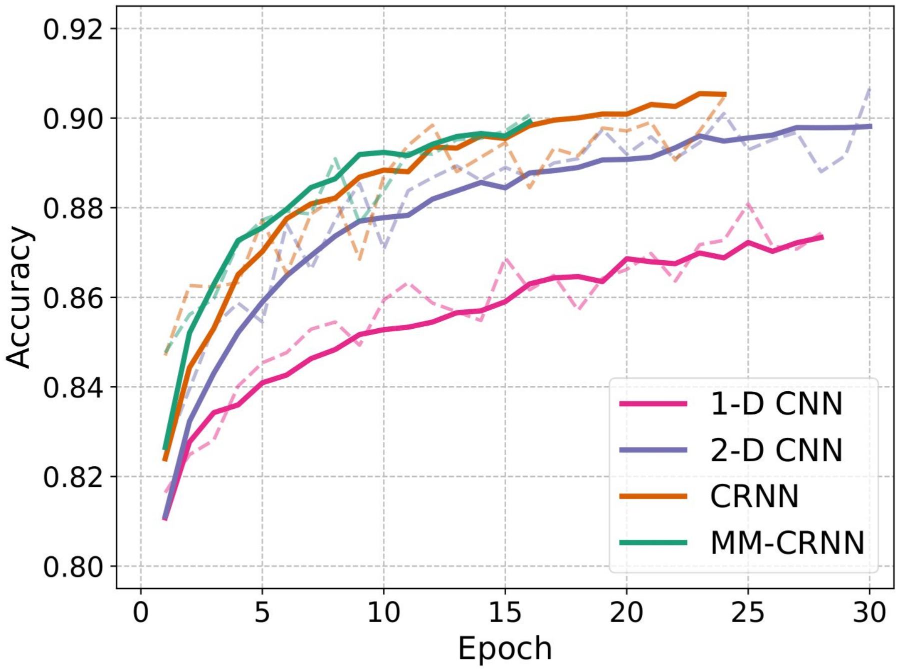
Conclusions

Models incorporating **LSTM** cells performed similarly to those using only CNN, but in fewer training epochs. This highlights the potential benefits of using recurrent-based methods in Music Genre Classification task.

Further research could explore the use of these models on **larger datasets** or attempt a **patch-wise approach** applied to audio tracks, allowing the use of even deeper architectures without being limited by the overfitting problem we faced in our research.

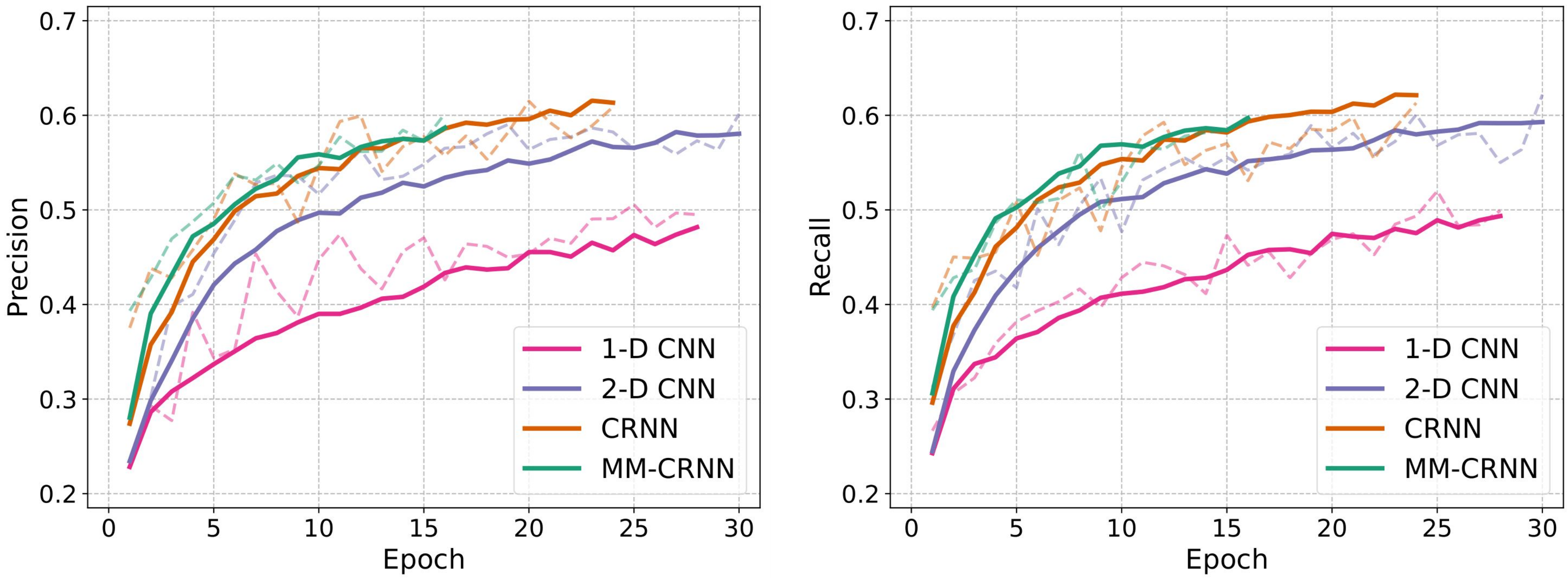
Training: Learning curves

Accuracy as a function of the training epochs:



Training: Learning curves

Precision and **recall** as a function of the training epochs:



Results: Confusion matrices

1-D CNN

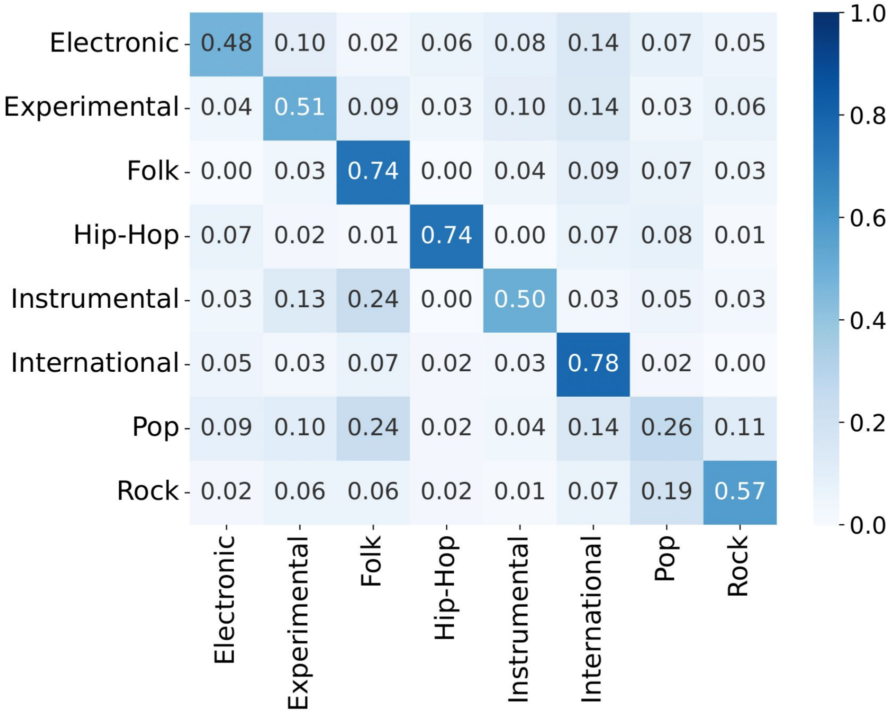
	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	0.44	0.08	0.02	0.17	0.07	0.07	0.06	0.08
Experimental	0.05	0.37	0.12	0.06	0.18	0.09	0.03	0.10
Folk	0.00	0.05	0.66	0.01	0.07	0.02	0.11	0.08
Hip-Hop	0.04	0.01	0.01	0.74	0.01	0.06	0.08	0.04
Instrumental	0.00	0.12	0.15	0.01	0.52	0.02	0.11	0.08
International	0.07	0.01	0.14	0.07	0.01	0.55	0.11	0.05
Pop	0.03	0.07	0.12	0.08	0.04	0.11	0.30	0.25
Rock	0.01	0.03	0.10	0.01	0.05	0.03	0.06	0.71

2-D CNN

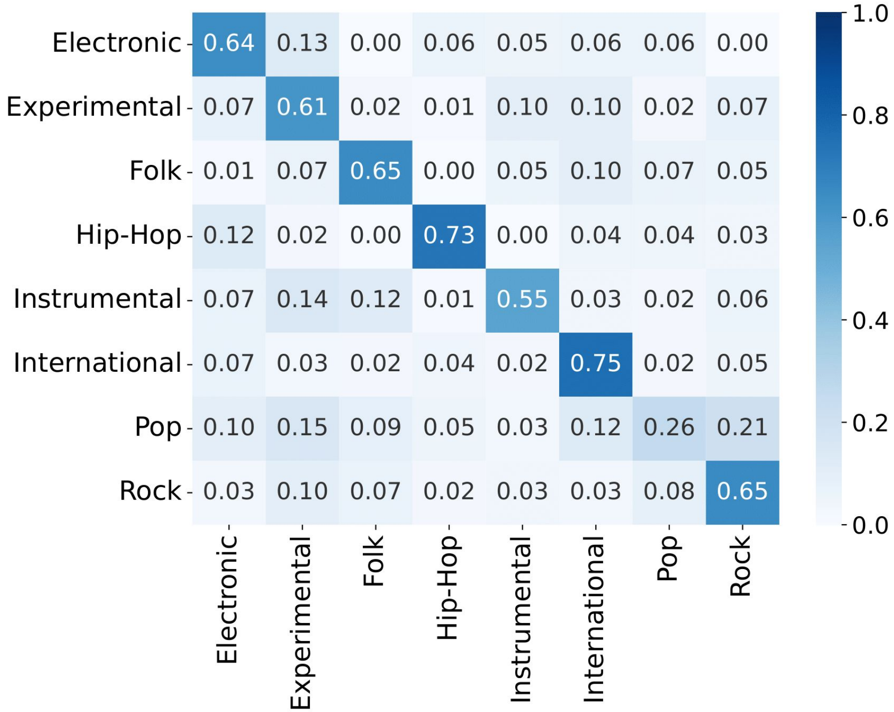
	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	0.52	0.15	0.00	0.19	0.04	0.04	0.04	0.02
Experimental	0.06	0.53	0.09	0.07	0.03	0.13	0.02	0.06
Folk	0.00	0.07	0.75	0.01	0.05	0.04	0.02	0.05
Hip-Hop	0.09	0.00	0.01	0.83	0.00	0.02	0.03	0.01
Instrumental	0.06	0.13	0.09	0.00	0.58	0.03	0.05	0.06
International	0.05	0.01	0.05	0.12	0.01	0.71	0.02	0.04
Pop	0.07	0.09	0.11	0.11	0.07	0.13	0.18	0.24
Rock	0.03	0.05	0.06	0.02	0.01	0.03	0.07	0.72

Results: Confusion matrices

RCNN



MM-RCNN



Equivariance vs. Invariance