Neural Music Classification - Project Report

Silvia Buccafusco
Polytechnic University of Turin
Turin, Italy
s290678@studenti.polito.it

Mehrnoosh Liravi
Polytechnic University of Turin
Turin, Italy
s289935@studenti.polito.it

Luca Francesco Rossi
Polytechnic University of Turin
Turin, Italy
s288386@studenti.polito.it

Abstract

Several studies have shown classical machine learning classification approaches applied to audio signals can lead to musical genre classification performances comparable with human competence. In this report, we present our solution applied to solve the Neural Music Classification project assigned in the Machine Learning and Deep Learning course at the Polytechnic University of Turin. We compare the classical feature based approach with the implementation of a Convolutional Recurrent Neural Network (CRNN), showing how a framework incorporating the temporal structure of audio signals combined with data augmentation can significantly outperform traditional solutions, achieving an accuracy of 86%.

1. Introduction and related work

Since music genres only represent a human attempt to create formal partitions of the huge existing music scene, music genre classification is a particularly challenging task, given that no sharp and clear boundaries between labels actually exist [1]. While in the literature there exist several attempts to solve Music Information Retrieval (MIR) problems with classical feature based approaches [1], [2], they required a deep domain knowledge as well as a huge handcraft work to obtain performances only comparable with human capabilities. However, thanks to modern deep learning architecture, we are able to efficiently emulate the neuroscientific ability of human minds of recognizing and classifying music genres and artists [3], [4]. To this purpose, Nasrullah et al. [4] presented what is likely the first comprehensive study of deep learning applied to music artist classification, implementing a convolutional recurrent neural network which outperforms in the optimal configuration previous attempts to such task taking advantages from the temporal structure of audio signal.
In this work, we compare the performances in classifying music genres of audio signals obtained applying our implementation of the classical feature based approach proposed in [1] and a CRNN. The work is organized in three main sections, namely a realization of the Musical Genre Classification according to [1], the PyTorch[1] implementation of the CRNN model proposed in [4] and lastly we present some strategies of data augmentation such as pitch shifting and windows overlapping which allow to reach the accuracy value of 86%.

2. Methodology

To the purpose of solving the music genres classification, we first propose our implementation of the classification based on the audio signals features extracted accordingly to [1] and then we outperform its results adapting the CRNN architecture presented in [4] to genre classification. It follows a description of both methodologies.

2.1. Feature based approach

Following the original work of Tzanetakis et al. [1], standard sets of features are extracted from each audio signal. To obtain the genres' classification, a multiclass support-vector machine classificator is applied.

2.2. Convolutional Recurrent Neural Network

Instead of summarizing the audio signals with low-dimensional handcrafted features, the advent of deep learning and the improvement of computational resources allow representing and classifying audio as spectrograms,

[1] https://pytorch.org/

a graphical representation of the signal frequency content over time.

ThE architecture adopted for genre classification, firstly proposed by [4] to the purpose of artists recognition, is shown in Figure 1. It grants to learn both the global and the temporal structure of input signals: several studies [5], [6] have indeed shown the efficacy of a recurrent unit in extrapolating the temporal structure with respect to other well-known architectures for audio classification.

3. Data preparation

In this section, after a brief description of the two dataset employed for results evaluation, the main steps performed to prepare raw signals both for features-based both for CRNN approach are described.

3.1. Dataset

Two datasets are used for evaluation, namely the artist20[2] and the GTZAN [1] dataset. While the former is only used to the purpose of evaluating the quality of the CRNN model replica, GTZAN is a well-known and widely-used resource for the music genre classification task. A summary of their content is provided in Table 1.

Property	artist20	GTZAN
Track length	30s	30s
Number of Tracks	1413	1000
Number of Artists/Genres	20	10
Album Per Artist/Genre	6	100
Bit rate	32 kbps	44.1 kbps
Sample Rate	16 kHz	22.05 kHz
Channels	Mono	Mono

Table 1. Description of artist20 and GTZAN dataset

3.2. Features-based approach data preparation

Three sets of features are extracted from the raw audio signal, namely the *timbral texture* (19 features), the *rhythmic content* (6 features) and the *pitch content* (5 features), resulting in 30-dimensional features vector. More details can be found in [1], [7]. The resulting dataset is finally rescaled between 0 and 1.

3.3. CRNN approach data preparation

Each raw audio is mapped to its spectogram. To this purpose, to each input signal is first applied the short-time Fourier Transform, defined as

$$STFT\{x(n)\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n},$$

where $x[n]$ and $w[n-m]$ represent the input digital signal and window function respectively, and then the squared magnitude of the result is taken.
Once spectrograms are created, frequency scale ($f = \frac{\omega}{2\pi}$) is transformed into Mel one (m) and then scaled in decibels (d) using the following equations:

$$m = 2595 \log_{10}(1 + \frac{f}{700}),$$

$$d = 10 \log_{10}(\frac{m}{r}),$$

where r is the reference power for log-scaling.

3.4. Data augmentation

Because of both the heterogeneous nature of songs when simply grouped together based on genre and their heterogeneous origins (analogue, digital, live, etc.), a good idea to improve the overall accuracy is data augmentation, which allows training and testing on examples obtained from the original ones by systematical transformations [8].
Following literature [8], [9] two data augmentation methods for the train set are adopted: *pitch shifting* and *slicing window overlapping*. A graphic description of their effects is provided in Figure 3.

For the former data augmentation technique, we shift the pitch of each song by half of the tone with SoX [3].
For the latter, since in the original version [4] songs are sliced with 3s non-overlapping windows, a 50% overlap (1.5s shift) allows creating entirely new spectrograms, thus allowing the model to learn more robustly.

4. Results

For feature-based approach, a ten-fold cross validation evaluation is used to repeatedly select 90% of GTZAN data for training and the remaining 10% for testing. The process is iterated 1000 times and the results, whose confusion matrix is displayed in Figure 4, are then averaged.
With a multiclass support-vector machine classificator, we reach an average accuracy of 67%, outperforming the original GTZAN result of 61%.

For CRNN model instead, 80% of data is used for training, 10% for validation and 10% for testing. The train-validation-test split is kept the same throughout the experiments, allowing a real comparison between models' performances.
When trained with the original GTZAN dataset, the reached accuracy is equal to 82%. However, training the model on the augmented train set, the results confirm that the CRNN learns more robustly as shown in Figure 5 and Figure 6, increasing its accuracy to 86% (with 3s slices, *song split* and *audio level* voting).

[2] https://labrosa.ee.columbia.edu/projects/artistid/

[3] http://sox.sourceforge.net/sox.html

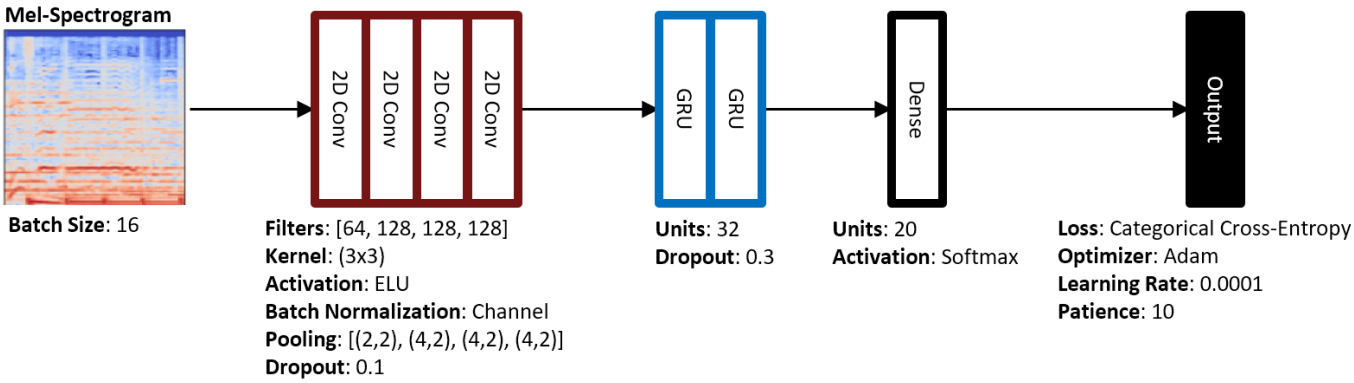

Figure 1. Model architecture as presented in the paper by Nasrullah et al. [4]

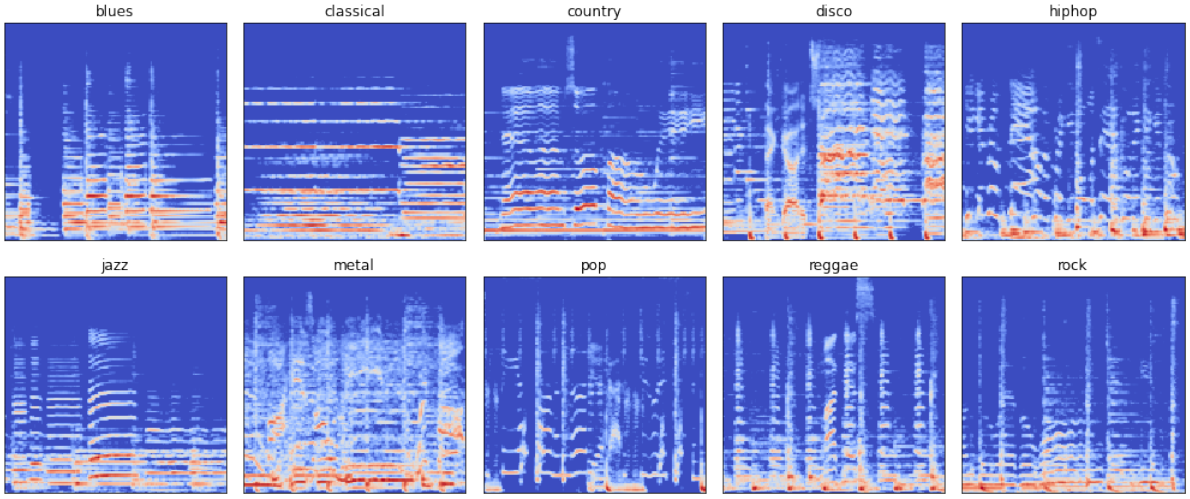

Figure 2. A random example of spectrogram for each genre from GTZAN

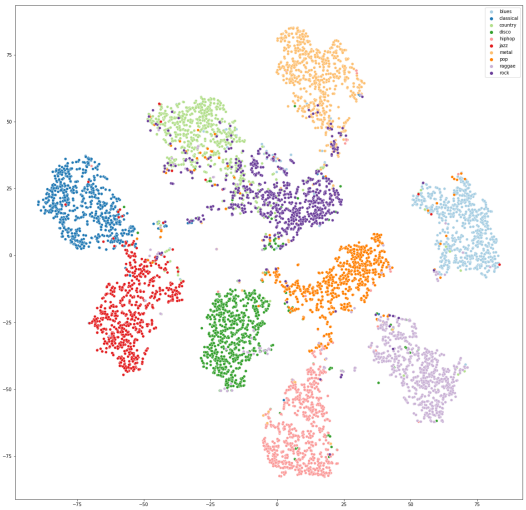

Figure 7. t-SNE representation of GTZAN dataset through the CRNN.

For this architecture, we present the t-SNE [10] representation of the GTZAN dataset through the last GRU, just before the last fully connected layer, a topological plot of data to get a graphical idea of how performing is the CRNN. It emerges that the genres are well discernible, with "rock" being the noisiest, probably due to the less-defined borders of this style with respect to the others taken into account in this study.

4.1. Method comparison

Comparing the results obtained in the carried out experiments in classifying the GTZAN dataset, we empirically demonstrate the better performance of CRNN approach with respect to classical features extraction. Even if we improve the accuracy of 61% reached by Tzanetakis and Cook in 2002 [1], we obtain an average accuracy of 67%, dramatically lower than the one reached by the CRNN ap-

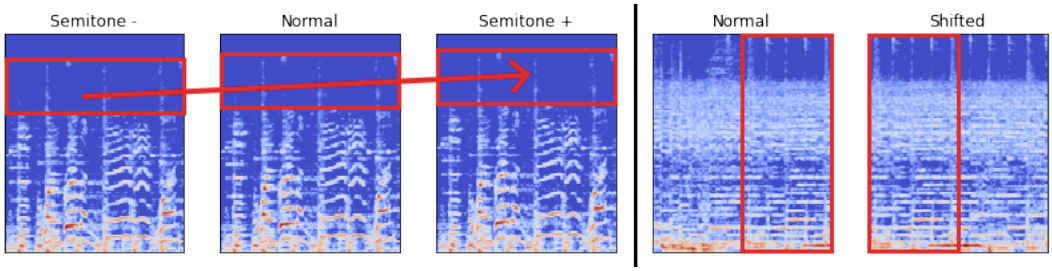

Figure 3. Semitone pitch shifting (left) and slicing window overlapping (right).

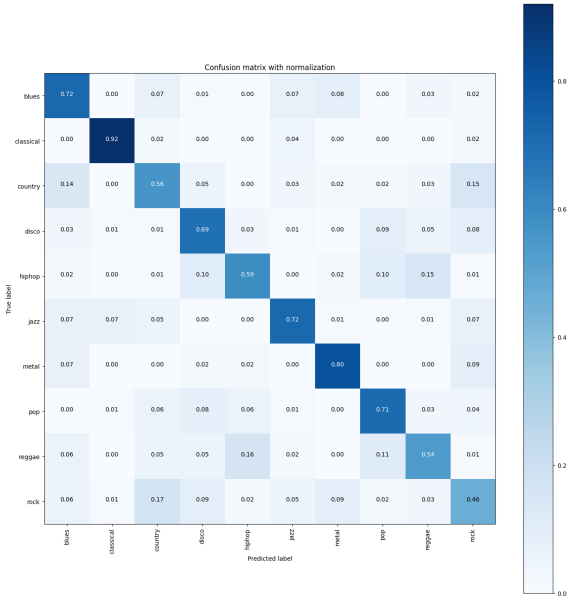

Figure 4. Confusion matrix from the feature-based approach.

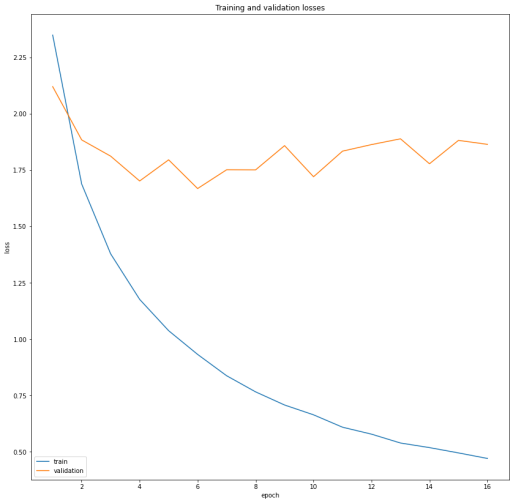

Figure 5. Train and validation losses with respect to the number of epochs for GTZAN dataset

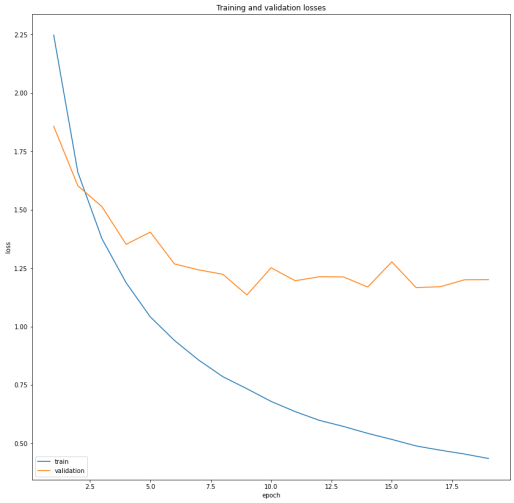

Figure 6. Train and validation losses with respect to the number of epoch for GTZAN augmented dataset.

proach[4] equal to 82%. This score can reach significantly higher values when the proposed architecture is combined with data augmentation, leading to 86% of correctly song genres identified on average.

5. Conclusions

In this work, we compared the performance of a classical feature-based approach for genre music classification with a Convolutional Recurrent Neural Network model. Even if we improved the accuracy of the GTZAN original paper from 61% to 67%, the higher value of 82% is obtained with original GTZAN dataset and this value reaches 86% when combining data augmentation with the proposed CRNN model. The experiments carried out also highlighted the importance of production for the model and therefore future works should move towards robustness in this area, especially for copyright detection application or recommendation systems.

[4]With 3s slice length, *song split* and *audio level* voting in order to obtain the best performance.

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, 2002.

[2] D. Ellis, "Classifying music audio with timbral and chroma features," in *ISMIR*, 2007.

[3] G. Kreutz and M. Lotze, "Neuroscience of music and emotion," 01 2007.

[4] Z. Nasrullah and Y. Zhao, "Music artist classification with convolutional recurrent neural networks," *International Joint Conference on Neural Networks*, 2019.

[5] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[6] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks." in *ISMIR*. Citeseer, 2013, pp. 335–340.

[7] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE transactions on speech and audio processing*, vol. 8, no. 6, pp. 708–716, 2000.

[8] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[9] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks." in *ISMIR*, 2015, pp. 121–126.

[10] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

A. Implementation of Music Artist Classification with CRNN

In this appendix, we provide details about our PyTorch implementation of the convolutional recurrent neural network model proposed by Nasrullah et al. in Music Artist Classification [4], showing that results coincide with the original ones.

Since the architecture is the same described in previous section, we only state results.

To evaluate performances, following the original paper procedure, two strategies of train-test split are applied to the artist20 dataset: the *album level*, meaning that one album for each artist is used as test set and another one as validation set, and the *song level*, meaning that 10% of songs are used as test set and another 10% as validation set (chosen randomly).

Once split, each song is sliced into audio parts of a given length (3s long clips turns out to be the best trade-off between training set size and audio length) and two strategies of performance evaluation are taken into account: the *frame level*, meaning that the f1-score is weighted among all slices, and the *audio level*, meaning that the f1-score is weighted among the entire songs, after having assigned to them the most frequent artist among their slices. Results are summarized in Table 2. As it is possible to notice, performance improves in the case of audio split, probably because of the variance reduction effect of voting leading to a more robust artist classification.

We also confirm the behavior highlighted by the authors [4]: better results are obtained when splitting by song instead of by album, likely due to the producer effect: test performance is much better when evaluating on unheard songs instead of unheard albums because the way an album is produced has to be considered meaningful and may also be a limited part of the artist's changeable style.

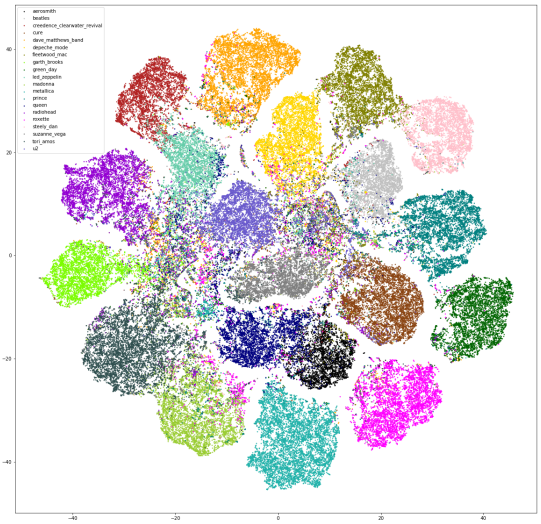

Figure 9. t-SNE representation of artist20 dataset through the CRNN.

Split level	weighted f1 frame	weighted f1 audio
Album	0.50	0.63
Song	0.72	0.93

Table 2. Results for CRNN model replica applied to artist20, coherent with the ones reported in [4]

As before, in Figure 9 we provide the t-SNE [10] representation of artist20 dataset to qualitatively evaluate how the network is performing. The artists turn out to be quite discernible, with some confusion just for the ones playing rock, for the reasons explained before.

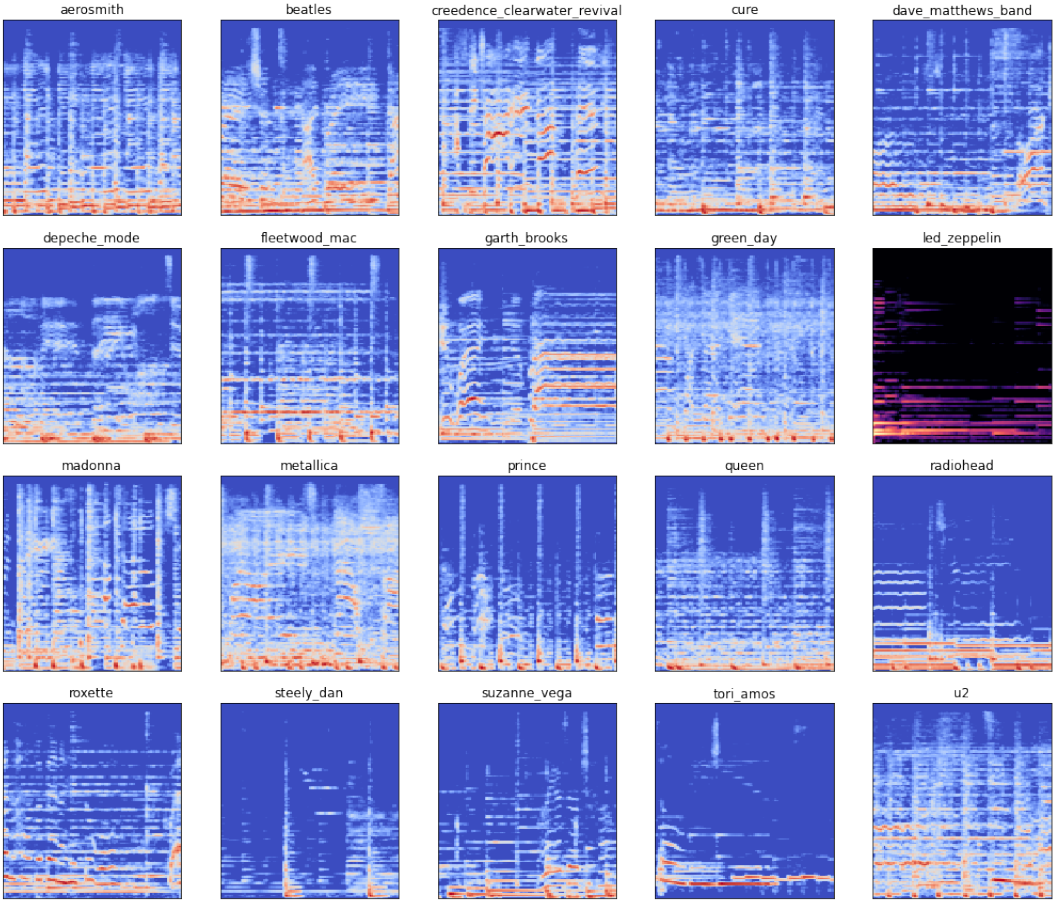

Figure 10. Random spectrograms from artist20 (one for each artist).

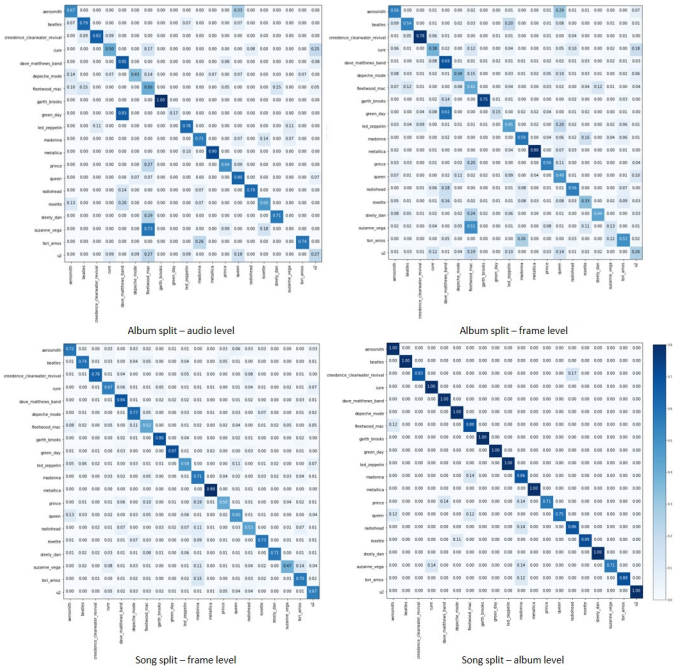

Figure 11. Confusion matrices for artist20 classification