Neural Music Genre Classification

Francesco Capobianco
Data Science
and Engineering
Politecnico di Torino
Email: s281307@studenti.polito.it

Alessia Leclercq
Data Science
and Engineering
Politecnico di Torino
Email: s291871@studenti.polito.it

Hafez Ghaemi
Data Science
and Engineering
Politecnico di Torino
Email: hafez.ghaemi@studenti.polito.it

***Abstract*—Conventional machine learning techniques in the field of music genre classification exploit tabular features, which are extracted using traditional signal processing tools from raw audio signals. Recently, deep-learning-oriented techniques, specifically convolutional neural networks, have been applied to time-frequency spectrograms instead. Moving in this way, music genre classification becomes a challenge in the field of computer vision. However, given the temporal nature of audio, the employment of recurrent units along with convolutional layers has proved to be useful for the concerned task. In this project, therefore, we extended an existing convolutional recurrent neural network (CRNN) architecture and evaluate it on one of the most famous benchmarks for music genre classification, the GTZAN dataset. Furthermore, using a pre-trained backbone network (ResNet) in a transfer learning setup, we achieved a state-of-the-art performance of 93.5% on the aforementioned benchmark. Finally, we tested our models on new songs unrelated to the GTZAN dataset, and analyzed the results.** [1]

I. Introduction

Music is one of the core aspects of human life, only in 2020 Spotify alone registered 2,015 million euros gross profit and spent 837 million euros on research and development [1]. Consequently, the need for efficient and high-performance music recommendation systems and other AI-based applications is central to the music industry, especially due to the availability and diffusion of music content online. One of the most important tasks in this field is music genre classification, which has become a popular pattern recognition problem in recent years.

Previous work mainly focused on the extraction of frequency-related and audio signal features, which do not consider the temporal structure of a song. Our approach, instead, is based on deep learning [2][3].

By performing time-frequency analysis and extracting spectrograms, both temporal and frequency contents can be managed simultaneously. Thus, we transformed music genre classification into a computer vision task. Since spectrograms also involve a temporal dimension, recurrent units such as gated recurrent units (GRUs) and long short-term memory (LSTMs) [4] can also prove to be useful.

Building upon the architecture proposed in [5] for artist classification, we are able to perform music genre classification and outperform traditional machine learning algorithms, such as support vector machine (SVM) and K-nearest neighbor (KNN). We trained our framework using different audio split lengths and compared the results. Finally, we pre-trained a backbone network (ResNet18) to perform transfer learning and observed performance improvements.

The sections of the report are structured as follows: In section II, an overview of the previous works in the field of artist and music genre classification is presented. In section III, the methods and architectures used and a brief description of the GTZAN dataset are provided. Section IV reports our evaluation results. A discussion around the results and the performance on different architectures and audio lengths, and performance of the models on external songs is given in section V. Finally, in section VI, a brief conclusion is provided.

II. Related works

Initial efforts in speech and music data analysis focused on speech recognition [6]. Later, the interest was extended to non-speech signals, for example, an interesting study was done to discriminate between music and normal speech [7]. Artist and music genre classification has recently gained attention among researchers [8][9][10][11]. In the pre-deep-learning era, most of the automated music genre classification algorithms were based on the extraction of feature vectors from raw soundtracks. However, genre classification presents two major complexities: on one hand, the definition of music genre is unclear [12][8], it mainly relates to a track's characteristics, e.g., instrumentation, rhythmic structure, harmonic content, and there exists a huge gap between the human and the automated performances in genre recognition [12]. On the other hand, the search for the right features to be extracted to correctly analyse and classify genres has been a hurdle for researchers. In [8] a framework for developing and extracting features for music content description and genre classification has been proposed. Among the multiple descriptive features for pattern recognition systems, [8] identifies timbral texture feature vectors, rhythmic content features, pitch content features, and real-time features. However, a limitation of this approach is the extended attention paid to frequency-related features and not to the inherent temporal and sequential aspects of music.

In [5] the spectrogram representation of songs has been used for artist classification. The power of a spectrogram consists of the opportunity to take advantage of the temporal aspects of the song, as well as keeping the frequency content. Moreover,

[1] The code is available on https://github.com/hafezgh/music_classification

spectrograms can be inverted to reconstruct the signal [9] [11]. The mel spectrograms are used as input images for a CRNN network to handle both the feature extraction stage and the sequential nature of music. The capability of a CNN to learn features for genre classification has been demonstrated in [9]. Thanks to the inverse transformation of deconvolved spectrograms into audio signals via the auralisation technique, Choi et. al. [9] demonstrated that on subsequent layers, the CNNs are able to identify patterns with an increasing complexity (from onsets to percussive and harmonics patterns), and that the learned features robustness to changes in key, chords and instruments increases as the number of layers increases.

Recurrent neural networks (RNNs) have demonstrated a great classification ability on tasks involving temporal data [6]. Choi et. al. [10] showed that employing gated recurrent units (GRUs) alongside convolutional layers improved the overall performance on genre classification. This is because the RNN structure is able to aggregate temporal patterns and maintain a global overview of the extracted features.

Even though the architecture in [5] works as the baseline for the proposed project, some differences can be observed with respect to the classification task. First, the prediction is performed on genres and not on artists. This mainly prevents some problems such as the artist's tendency to vary in style, and also the producer effect, which have been acknowledged to have a strong impact on the artist classification task. Yet, since music genre classification strictly depends on the song's characteristics, the extracted features (e.g. the instrumental identification, and also the higher abstract patterns) are expected to be meaningful in terms of recognizing the genre, and we expect the presented architecture to achieve high performance. Secondly, we employed the GTZAN music genre dataset for evaluation and not the artist20 dataset used in [5] whose labels refer to only artists.

Transfer learning [13] has been proven to improve classification performance on small image datasets, since the network weights are already tuned to perform well on a large-scale dataset such as ImageNet [14]. A famous network used as the backbone network in computer vision tasks is ResNet [15] and its different variants. ResNet utilizes residual (skip) connections to alleviate the problem of vanishing gradients which arises when the network becomes too deep. In a transfer learning setup, we assume that the network is able to detect features such as the vertical and horizontal lines described in [9], and because of the network's depth, it should also be able to recognize more complicated patterns (e.g., the harmonic ones) in the last layers. Therefore, also the limitation of small music genre datasets in terms of number of samples can be overcome using transfer learning.

III. Methods

1) Dataset: The GTZAN dataset, published in 2002 [8], is the most-used public dataset available for music genre classification. The popularity of GTZAN increased in recent years, only in 2010, 2011 and 2012 it has been cited and used in 65, 55 and 47 articles respectively [16]. Its content is divided into 10 different genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock. Each of them consists of 100 30-seconds length soundtracks resulting in a total of 8-hour audio data. The files were collected according to a variety of different recording technologies and sources, such as CDs, radio, microphone. Table I gives a descriptive summary of the GTZAN dataset.

In the version we employed, along with the original .wav songs, the images folder containing the image spectrograms of the tracks was already provided, as well as two csv files containing extracted spectral features for both 30-seconds and 3-seconds length audio chunks. The song's sample rate is 22050 Hz and the provided spectrograms are 3-channel 288x432 RGB images.

The train-val-test split to train our deep-learning frameworks is performed by randomly dividing the original dataset in 80% training, 10% validation and the remaining 10% for testing.

The two .csv files contain frequency-related features that have been used to train and test the standard machine learning algorithms for supervised multi-classification problems. In this case, we only used the provided 30-seconds and 3-seconds length audio features and split them randomly in 80/20 training and test sets. An example of the spectrograms from GTZAN is provided in Figure 1

TABLE I
Characteristics of the GTZAN Dataset

Total Number of Tracks	1000
Track per genre	100
Number of genres	10
Sample rate	22050 Hz
Channels	Mono

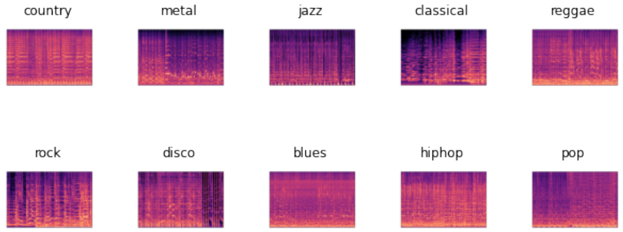

Fig. 1. 30-second spectrograms from the GTZAN dataset

2) Audio preprocessing and data augmentation: Although the thirty-second spectrograms were already provided, we still needed to implement a function for producing mel-spectrograms (The librosa python package [17] was used for generating spectrograms). Producing the spectrograms was done to augment the limited amount of samples (only 1000) present in the dataset and improve deep learning performance, and also to replicate data slicing done in [5] to find the best possible audio length split in terms of performance. We used 30s, 10s, 3s and 1s length split for training and evaluating our neural networks, while the standard machine learning algorithms have only been trained using the 30 and 3-seconds

split feature vectors.
The implemented function exploits the STFT function in (1) to create the spectrogram and then transforms it into the mel scale in (2) before scaling it into decibels using (3) [5]. A summary of the parameters used for this process can be found in Table II.

$$STFT\{x(t)\}(\tau, w) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-iwt}dt \qquad (1)$$

$$m = 2595 log_{10}(1 + f/700) \qquad (2)$$

$$d = 10 log_{10}(m/r) \qquad (3)$$

TABLE II
PARAMETERS FOR PRODUCING THE MEL-SPECTROGRAMS USING STFT

Sampling Rate	22.5kHz
Number of Mel Bins	192
Highest Frequency	8kHz
Hop Length	67

3) Evaluation Metrics: For evaluation purposes, we decided to measure the F1-score and accuracy across all samples in the test set. According to [5], the F1-score is helpful to reduce class imbalance from the variance of song lengths when each frame is considered as an independent one. However, the original content of the GTZAN is already balanced in terms of quantity of songs per category and song's length (all the original .wav are 30-sec length). As a consequence, accuracy or F1-score are almost the same when evaluating this dataset, and one of them is sufficient for evaluating model performance. So, the F1-score is reported to be consistent with the evaluation performed in [5].
Accuracy on training, test and validation sets is computed for each epoch as the number of correctly predicted labels divided by the total number of predictions, while the F1-score (4) is computed considering the global number of true-positives (5), false-positives (6) and false negatives.

$$F1 = \frac{2 * (precision * recall)}{(precision + recall)} \qquad (4)$$

$$precision = \frac{truePositive}{(truePositive + falsePositive)} \qquad (5)$$

$$recall = \frac{truePositive}{(truePositive + falseNegative)} \qquad (6)$$

4) Base architecture: As the base architecture, we implemented the network model used in [5] for music artist classification (Fig. 2). This model consists of a CRNN in which the CNN part is able to detect frequency features in the songs spectrograms and the GRU units maintain the focus on the temporal sequence of those patterns. The architecture presents four 2D-convolutional layers for feature detection with 3x3 kernels and an ELU activation function. Also, batch normalization and dropout (ratio= 0.1) are performed on the channel dimension as regularization technique. The output of the CNN is fed into two 32-units GRU units, which have been added to the model to capture the temporal dimension of the extracted features, and in this scenario opted for GRUs instead of LSTMs since they have similar performances, but less parameters to train and a simpler internal structure. Finally, the network ends with a 10-unit dense layer to assign probabilities to each class using softmax activation function. The chosen optimizer is Adam with a 0.0001 learning rate (the original learning rate used was 0.001 which was reduced for training stability). The loss function used is categorical cross-entropy because of the multi classification nature of the music genre classification problem.
Also, early stopping with patience of 10 epochs is employed to shorten the training procedure and avoid overfitting. Globally, the baseline network presents 372,048 trainable parameters. A summary of this network's architecture and hyperparameters can be found in Table III.

TABLE III
CHARACTERISTICS OF THE ARCHITECTURES

Hyperparameters CNN	Base Architecture	Extended Architecture
Filters	[64, 128, 128, 128]	[64, 128, 256, 512, 512]
Kernel	(3x3)	(3x3)
Activation	ELU	ELU
Batch Normalization	Channel	Channel
Pooling	[(2,2), (4,2), (4,2), (4,2)]	[(2,2), (2,2), (2,2), (4,1), (4,1)]
Dropout	0.1	0.1

Hyperparameters	Recurrent and Dense layers
Recurrent Layers Units	32
Dropout	0.3
Dense Layer Units	20
Dense Layer Activation	Softmax

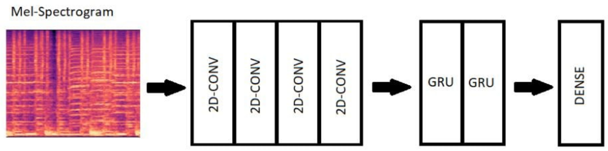

Fig. 2. Base architecture proposed by [5].

5) Extended Architectures and Transfer Learning : The base network has been extended by adding one additional 2D-convolutional layer with larger number of filters, while keeping the same RNN structure and dense layer (see Table III for architecture details and Figure 3 for graphical summary of this extension). Adding an additional layer enables the extraction of more complex and high-level features. Also, increasing the number of filters increases the diversity of the extracted patterns. In [9], the authors demonstrated that the fifth convolutional layer in the network can detect harmo-rhythmic structures and overlapping harmonic patterns, which are difficult to interpret, but they might be helpful for identifying the genre. Because of the introduced modifications also the number of lernable parameters increased to a total of 3,914,064 trainable parameters.

As another extension to the base architecture, we employ transfer learning to further improve classification performance. ResNet18 [15] replaced the original CNN structure. Among the 10 convolutional layers of ResNet, we locked the pre-trained weights (on the ImageNet [14]) in the first six layers, but left the last four to be trained on the spectrograms. This is justifiable because the low-level features, which are detected in the first convolutional layers, are common between every image dataset. However, the more abstract and high-level features vary in different datasets. We also increased the number of recurrent units (32-unit GRUs) to three to better capture the temporal information. A total of 58,144,138 parameters are present in the network. See Figure 4 for the graphical summary of the ResNet-based architecture.

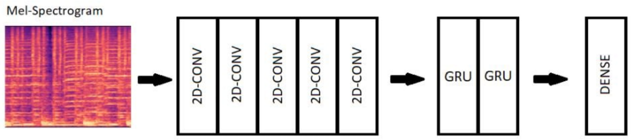

Fig. 3. Base network extended by adding one additional 2D-convolutional layer with a larger number of filters while keeping the same RNN structure and dense layer.

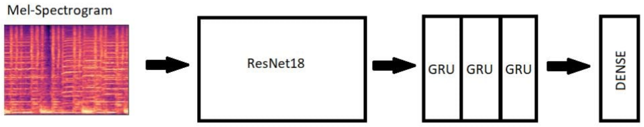

Fig. 4. A transfer-learning based extension to the base architecture, where a pre-trained ResNet18 [15] replaced the original CNN structure.

IV. Results

The classification performance for traditional machine learning algorithms are reported in Table IV. Four standard machine learning algorithms have been tested on the 30-seconds and 3-seconds features. Among these algorithms, we also tuned the penalty hyperparameter (C) of the SVM and the number of neighbors in KNN (tested with k=1 to 20). The best performance on the test set has been achieved using SVM and on the feature vectors of 3-second chunks (F1-score = 0.9161). This performance can be considered a baseline to be outperformed by one of our neural network-based extensions. The three deep-learning-based architectures are trained using four different training sets (30-sec, 10-sec, 3-sec, and 1-sec audio chunks). Table V presents the performance of these architectures. The best performance is achieved using the ResNet-18 transfer-learning-based approach on the 1-sec split length dataset. The accuracy-per class and the confusion matrix for the test set for the best-performing model are plotted in Figure 5 and Figure 7.

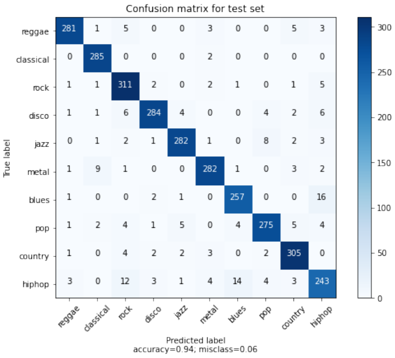

Fig. 5. Confusion matrix for test set

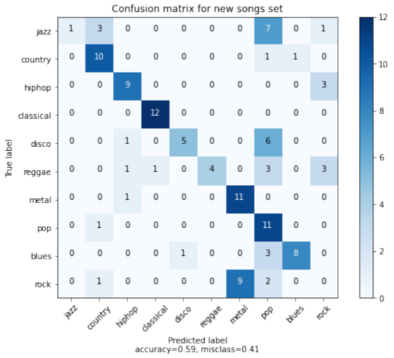

Fig. 6. Confusion matrix for test on our songs

V. Discussion

1) Convergence analysis: As can be seen from the results, even after applying multiple regularization techniques, a variance between the train and validation/test accuracies is present. When we are building a predictive model, we care about how it performs on unseen data. While the gap between training and validation can be a useful heuristic sometimes, it does not always mean overfitting. With a sufficiently complex model we always expect a gap between training and validation. The training performance starts to matter only when it comes at the expense of a worsening validation score. That is, when we are memorizing the training data and performing worse on unseen data as a result. But here, the generalization is not affected by the high training accuracy, and we still obtain an acceptable performance on the validation set.

The number of epochs needed for convergence of the three proposed deep learning models for each training set is given in Table V. It can be observed that the number of epochs needed for convergence has no apparent correlation with the complexity of the model or the number of training data. This may be attributed to the use of early stopping.

The convergence diagrams for the best-performing model (transfer-learning-based) given in Figure 7 show the loss decrease and train and validation accuracy increase through the training process (number of epochs). It can be seen that the early stopping has been triggered when no decrease in the

TABLE IV
TRADITIONAL MACHINE LEARNING APPROACHES RESULTS

	Train F1 score 30 sec. (%)	Test F1 score 30 sec. (%)	Train F1 score 3 sec. (%)	Test F1 score 3 sec. (%)
SVM (default)	88.89	69.63	92.23	85.98
Tuned SVM (C = 10)	99.87	78.03	99.65	**91.61**
KNN (k = 1)	1	66.67	99.89	91.47
KNN (k = 5)	78.75	69.41	93.44	89.67
Random Forest (default)	100	68.04	100	87.65
Logistic Regression (default)	100	67	100	72.89

TABLE V
RESULTS OF THE DIFFERENT ARCHITECTURES

Model (split-length)	Train F1 (%)	Validation F1 (%)	Test F1 (%)	New Songs Test	#Epochs before early stopping
Base(30)	99.38	57	69	30	131
Extended (30)	54.43	55.37	55.08	45	208
Transfer (30)	99.88	71	88	57.5	136
Base (10)	99.71	77.52	76.72	46.67	251
Extended (10)	99.8	78.83	80.33	42.5	98
Transfer (10)	99.92	89.25	89.18	59.17	65
Base (3)	99.26	89.25	89.18	59.17	65
Extended (3)	99.84	90.7	90.6	47.5	197
Transfer (3)	99.92	89.25	89.18	44	65
Base (1)	99.34	90.93	91.7	49.5	224
Extended (1)	99.76	90.17	89.74	43.75	149
Transfer (1)	99.92	93.5	93.5	46.5	162

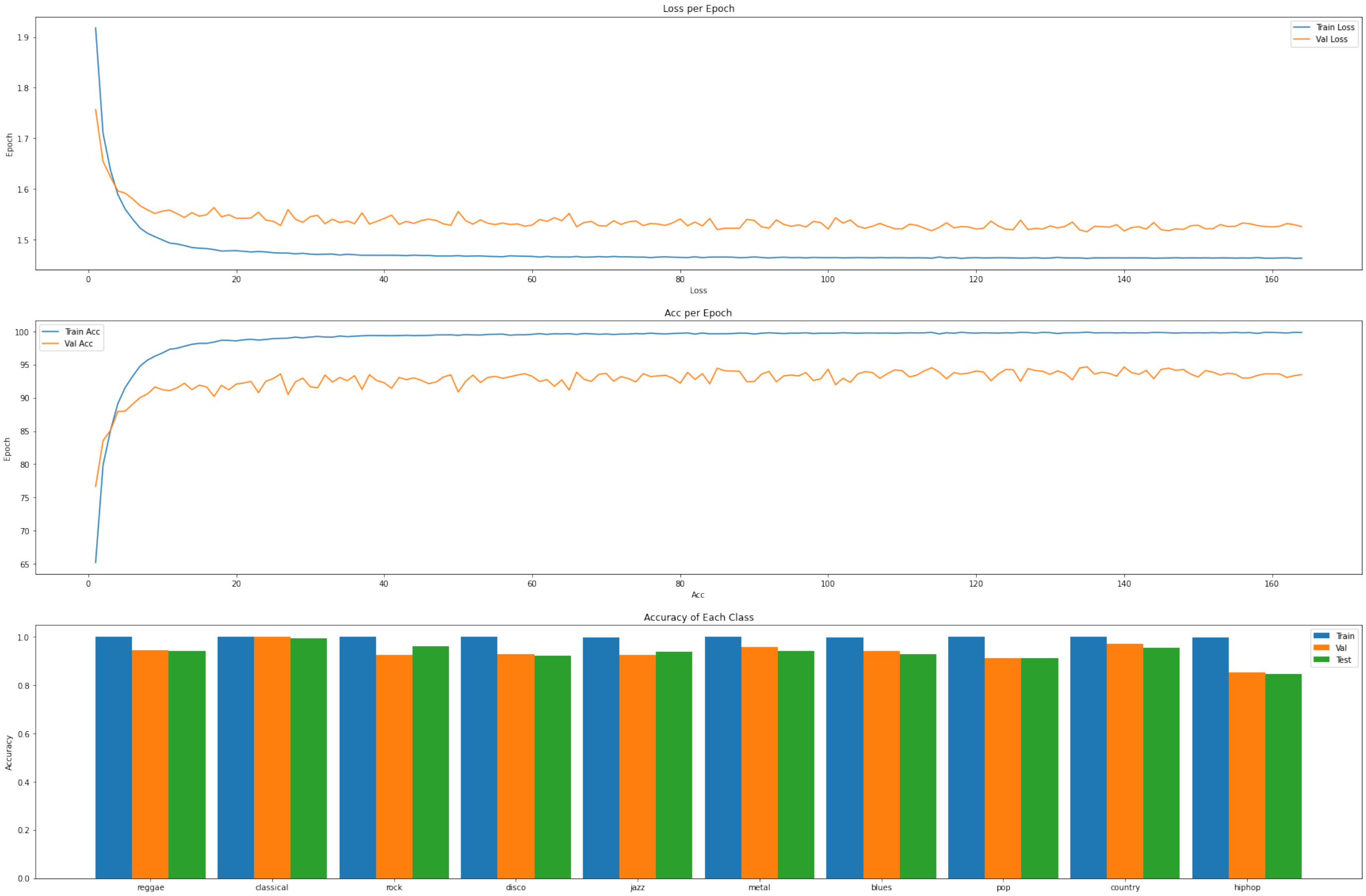

Fig. 7. Accuracy, Loss and Accuracy per class

validation loss is observed for ten consecutive epochs.

2) The possible trade-off between the number of samples and the length of the splits: Generally, we can say that longer chunks contain more temporal information, while shorter chunks allow the network to be trained on more data. A trade-off between data availability and maintaining temporal information may be found when we modify the split length. This trade-off is present in the work of Nasrullah et. al. [5] where using 3-sec chunks achieve a better performance compared to one-sec chunks, although one-sec chunks provide more training data. In our case, the best performance is achieved using the one-second chunks. This may be justified, by the fact that, recognizing artists using one-sec chunks may be impossible, but, recognizing genre may be possible. So, the trade-off in our case, is in favor of more training data, rather than maintaining more temporal information

3) Results on external songs (data samples) : Ten songs, one per each class, and not included in the dataset were chosen, trimmed to two minutes length and split into 30-sec chunks. After extracting spectrograms with different split lengths, the samples were classified using the corresponding ResNet-based model for each split length. The confusion matrices plotted in Figure 6 show the classification performance. The lower performance compared to the test set may be attributed to the lack of data and poor variety of songs in the GTZAN dataset which does not allow the model to accurately capture each genre's feature patterns. The external songs may include feature patterns not present in the dataset tracks for a specific genre.

4) Further works and limitations: The first limitation is related to the dataset used, the GTZAN. As it can be seen from the results obtained on the external songs, the GTZAN dataset might not be diverse enough for capturing each genre's feature patterns. Augmenting this dataset using external labeled songs may improve the generalization abilities of the trained models. The author of [16] mentions that the GTZAN dataset contains repeated musical patterns among each genre. Therefore, additional samples or training on datasets other than GTZAN could be useful to boost the performance.

A second limitation is related to the model complexity with respect to the computational power. The availability of higher processing capacity is crucial to be able to train deeper and more complex models, which might correspond to the detection of more complex, but useful patterns for genre classification.

Finally, according to the results of this work, transfer learning is a useful technique for music genre classification and further works could focus on the use of other backbone networks other than ResNet18 for pre-training.

VI. Conclusion

In this project, we built on a previous architecture, a CRNN network used for music artist classification. We extended this architecture, expanding it by adding another convolutional layer and using more filters, and also modifying it to be compatible with transfer learning. Afterwards, the architectures were used for music genre classification using mel-spectrograms as the training data. Specifically, the final network that used ResNet-18 as its pre-trained backbone was able to reach the state-of-the-art performance on the GTZAN, outperforming previous works and the conventional machine learning algorithms. The best accuracy achieved on the test set is 93.5%. Finally, we investigated the effect of different audio split lengths and examined the possible trade-off between the preservation of the temporal structure and the number of data samples available for training and tested our models on songs external to the dataset.

References

[1] H. Gutierrez, Annual Report Spotify 2020, Tech. rep. (2020).

[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.
URL http://www.nature.com/articles/nature14539

[3] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[4] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[5] Z. Nasrullah, Y. Zhao, Music artist classification with convolutional recurrent neural networks, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[6] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE international conference on acoustics, speech and signal processing, Ieee, 2013, pp. 6645–6649.

[7] J. Saunders, Real-time discrimination of broadcast speech/music, in: 1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings, Vol. 2, IEEE, 1996, pp. 993–996.

[8] G. Tzanetakis, P. Cook, Musical genre classification of audio signals, IEEE Transactions on speech and audio processing 10 (5) (2002) 293–302.

[9] K. Choi, G. Fazekas, M. Sandler, Explaining deep convolutional neural networks on music classification, arXiv preprint arXiv:1607.02444 (2016).

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

[11] H. Deshpande, R. Singh, U. Nam, Classification of music signals in the visual domain, in: Proceedings of the COST-G6 conference on digital audio effects, Vol. 1, Citeseer, 2001, pp. 1–4.

[12] S. Lippens, J.-P. Martens, T. De Mulder, A comparison of human and automatic musical genre classification, in: 2004 IEEE international conference on acoustics, speech, and signal processing, Vol. 4, IEEE, 2004, pp. iv–iv.

[13] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: A survey, IEEE transactions on neural networks and learning systems 26 (5) (2014) 1019–1034.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[16] B. L. Sturm, The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use, arXiv preprint arXiv:1306.1461 (2013).

[17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python, in: Proceedings of the 14th python in science conference, Vol. 8, Citeseer, 2015, pp. 18–25.