Multi-head music genre classifier

Agostino Aiezzo - 982514

July 16, 2023

1 Introduction

The aim of this project is to build a Convolutional Neural Network which is able to correctly classify the genre of a music track given a representative image of the track itself. The reason behind that is to provide further proofs that Computer Vision and Machine Learning can be applied to unusual scopes. In addition, there's not so much material about this kind of task in current literature. This report is organized as follows: in Section 2 you'll find a series of preliminary actions without which it would not have been possible to complete the project; in Section 3 there's a description of the adopted model and the difficulties in the early stages of design; subsequently, in Section 4 performances of the model will be analyzed in terms of loss and confusion matrix; eventually, in Section 5 final considerations about the project and future developments are investigated.

2 Before proceeding

As mentioned in Section 1, before proceeding with the main part of this project, some preliminary actions are necessary in order to make everything works. First of all, the GTZAN dataset is used: this is the most diffused public dataset for evaluation in machine listening research for music genre recognition (MGR). It consists of 1000 tracks of 30 seconds each which can belong to a single genre selected from 10 possible values (blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock). However, it's been recently retired from the list of torchaudio's dataset probably because of some faults as mentioned in [1]. Nonetheless, a *custom dataset* using the original soundtracks folder retrieved on Kaggle [2] was created (just one audio data has been removed because it was buggy). Also, a couple of utils functions have been defined in order to manage the annotations (the labels for each track) and the creation of images. About this latter, several attempts with many kind of depictions of different duration have been made in Section 3, that is about the definition of the overall architecture.

3 The model

The basic idea is to take and audio track, generate an image from it and give it in input to a CNN in order to obtain the label that is the genre of the soundtrack. In particular, the representation initially used for each audio data was the *Chromagram* (Figure 1), a powerful tool for analyzing music according to its pitches. One important property of chroma features is that they capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation [3]. The main reason why chromagram was chosen is to find and test an alternative transformation to the omnipresent spectrogram that is normally used when audio is involved in CV applications.

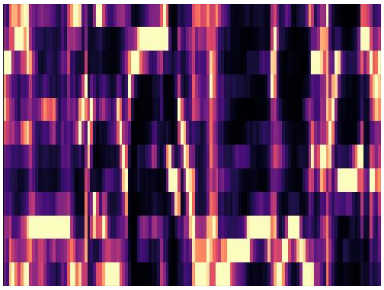


Figure 1: Example of a Chromagram of a blues music track

For what concerns the model, inspired by VGG, this is a simple CNN with 5 convolution layers each followed by a Relu and a MaxPool; 5 FC layers conclude the architecture. For training, a *polynomial* learning rate scheduler of second order is adopted to make training more stable and effective. Once trained, the model immediately showed to be affected by *overfitting* as can be observed in Figure 2. Also, accuracy was averagely around 30-40%.

Obviously, this is due to the low number of data at disposal. Even trying with some dropout, it was not possible to reach satisfying results. As a consequence, instead of creating images on the whole 30 secs audio track, the chromagram was created taking sound every 3 secs, obtaining 10 images for each music track. Despite the higher number of data, accuracy performances were still low (around 50-55%) and overfitting occurs anyway. Other fractions to partition the soundtracks were considered (every 1,

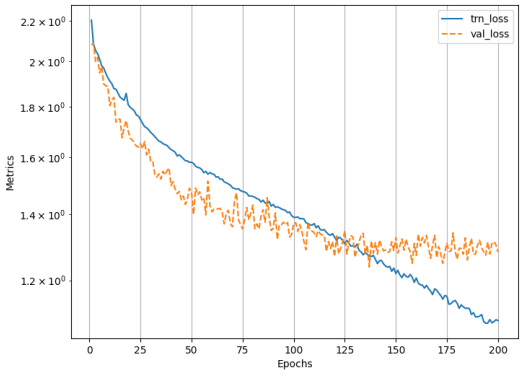


Figure 2: Overfitting of the model due to low number of data entries

$\frac{1}{2}$, $\frac{1}{10}$ of sec) obtaining more and more data every time. However, the situation still remained the same. So, the problem had to be somewhere else: if the cause was not the number of images, maybe it was the number of features. This hypothesis was sustained by the correspondent confusion matrix, as shown in Figure 3, where it's possible to notice that the model had difficulty to distinguish certain genres from the others.

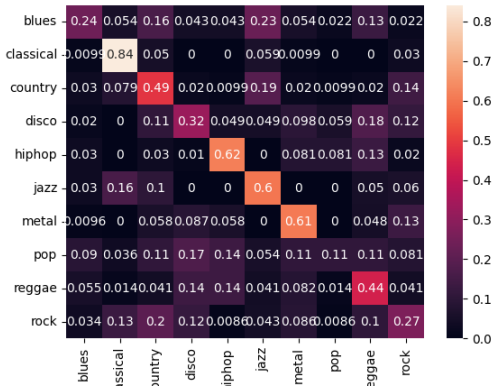
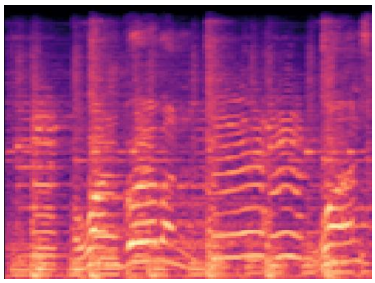


Figure 3: Confusion matrix, at this point of the model, shows that some genres are not well distinguished from others

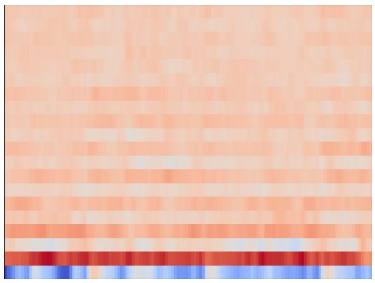
So, in order to increase the characteristics at disposal of the model, the idea was to use other image representations of the same audio track, combine them and obtain the output label. Consequently, two other kind of depictions were considered:

- the *Mel spectrogram* (Figure 4a), that is basically a spectrogram in mel-scale (a perceptual scale of pitches judged by listeners to be equal in distance from one another) [4]
- the *Mel-frequency cepstral coefficients (MFCCs)* (Figure 4b), that are coefficients that collectively make up an MFC (a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency) [5].

In order to manage this multiple input, the architecture used so far was updated. The idea was to take the convolutional layers of the previous version of the model and define three heads, one for each type of image. Afterwards, once each head has elaborated its correspondent input, the associated output features maps are concatenated and given in input to the FC layers for the classification. For what concerns the number of layers, channels, filter sizes, and so on, they remained pretty the same as before. The only extra addition at this point is batch normalization that is used to improve stabilization and convergence. As Section 4 will demonstrate, this peculiar configuration of the model is able to achieve satisfying results.



(a) Examples of Mel spectogram of a blues track

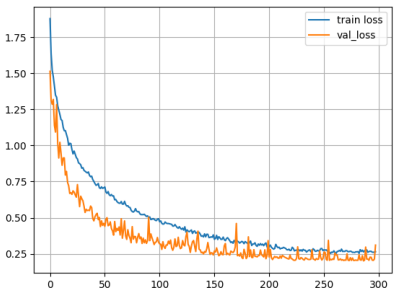


(b) Example of MFCCs of a blues track

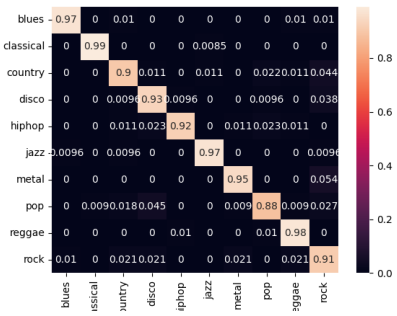
Figure 4: The other two image representations used in combination with Chromagram

4 Results

As Figure 5a shows, the loss has been drastically reduced with the multi-head approach. The shape is good, quite stable and little variable. Overfitting still occurs, but to solve this problem, once and for all, at this point, an *early stopping* mechanism was implemented. For what concerns the confusion matrix in Figure5b, all the classifications surpass 90% of accuracy except for the "pop" label: a penalization term into the loss was used trying to squeeze the last drop of performance, but it didn't work so well. In any case, results are now very satisfying with 94% of accuracy achieved.



(a) Training and Validation loss of the multi-head genre classifier



(b) Confusion matrix of the multi-head genre classifier

Figure 5: Final loss and confusion matrix of the multi-head genre classifier

5 Conclusions

It's been really complicated to find the right approach to solve this problem: it was not a obvious solution to think about multiple inputs and then combine them and so on. About this, also alternative approaches to merge features maps from each head were tested (average and sum), but relevant differences have not been observed. Also, other kinds of images (spectral contrast, tonnetz) and, consequently, more heads have been tried, but performances did not vary. A lot of time has been lost to set all the parameters (which could actually be a pretext to implement a grid search in the future) and to find the right learning rate scheduler. In general, also a more efficient management of the dataset would be necessary. Hardware has been a limit on certain occasions, but actually the problem is that the network has 5 millions parameters that could be further refined. Training time also was a problem and, in fact, this is why learning rates schedulers, batch normalization and a mechanism that constantly saves the model were adopted. At the end of this project, the achieved results can be considered good, still improvable, of course, but this work is the demonstration that machine learning and computer vision can be applied, with smart solutions, to fields that apparently do not share nothing with vision.

References

[1] Bob L. Sturm. The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *ArXiv*, abs/1306.1461, 2013.

[2] Andrada. Gtzan dataset - music genre classification, 2020. URL `https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification`.

[3] Wikipedia. Chroma feature, 2022. URL `https://en.wikipedia.org/wiki/Chroma_feature`.

[4] Wikipedia. Mel scale, 2023. URL `https://en.wikipedia.org/wiki/Mel_scale`.

[5] Wikipedia. Mel-frequency cepstrum, 2023. URL `https://en.wikipedia.org/wiki/Mel-frequency_cepstrum`.