POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Musical Genre Recognition

Numerical Analysis for Machine Learning Project

Authors: **Stefano Arcaro**
Gianmarco D'Onofrio

Academic Year: 2022-2023

Contents

1 | Introduction

In the literature, there exist various datasets concerning the task of music genre recognition (MGR). Although quite small, the GTZAN public dataset represents a benchmark in academia, and is widely used.

Among the many published works, the best-performing models are described by Liu et al. [1], with an accuracy score of 93.65%, just a bit less than the previous state-of-the-art performance, achieved by Liu et al. [2] with their Bottom-up Broadcast Neural Network (BBNN) model with a 93.7% classification accuracy.

The aim of this project was to re-implement the work done by Liu et al. [1] on a middle-level learning feature interaction (MLFI) method with deep learning. The task of this paper was to explore the interaction between learning features of middle layers and its impact on the classification results of the whole model.

This report documents the findings of our project for the Numerical Analysis for Machine Learning course.

2 | Data pre-processing

Liu et al. [1] proposed several models, all of which made use of both visual and audio features. All these features correspond to a 3-second audio file.

2.1. Visual features

Given that humans are better at distinguishing sounds in the low-frequency domain than the high-frequency one, using a perceptual scale to describe the music input data visually is more reasonable compared to a standard logarithmic scale (spectrogram).

The Mel scale is exactly that: a perceptual scale of pitches judged by listeners to be equal in distance from one to another. Figure 2.1 illustrates the difference between a spectrogram and a Mel spectrogram.

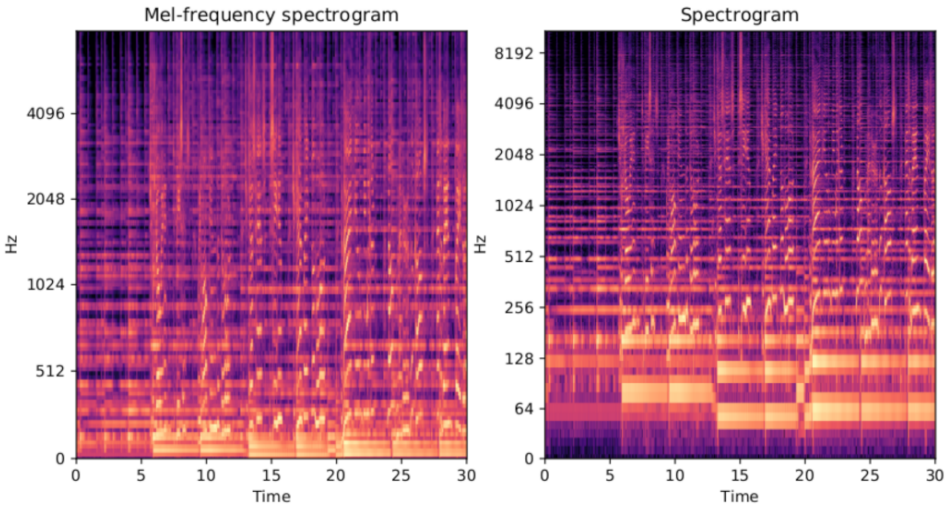

Figure 2.1: Mel-frequency spectrogram compared to the spectrogram.

We used Librosa to extract the Mel spectrograms with 128 Mel-filters, a frame length of 2048 and a hop size of 512.

2.2. Audio features

They chose nine audio features: all of them but the tempo are described in the form of mean and variance. The following table shows the chosen features:

Timbral Texture Features	Other Features 1	Other Features 2
Chroma	Harmonic Component	Tempo
Root Mean Square (RMS)	Percussive Component	
Spectral Centroid		
Spectral Rolloff		
Zero-Crossing Rate (ZCR)		
MFCCs (1-20)		

Table 2.1: The set of audio features.

With the GTZAN dataset the audio features for the 3-sec segments are already computed and provided in a csv table.
We tried to perform live prediction on arbitrary music, so we had to compute the audio features ourselves, and to achieve this we used Librosa.

2.2.1. What we did differently

While analyzing the paper [1] we had to re-implement, we noticed an important step in the pre-processing of the data was being left out, as we could not find any mention of it: this step being the normalization of the input data, both visual and audio features, before feeding it to the models.

As we believe normalization to be a crucial step for Deep Learning techniques, we decided to implement it. This simple step alone helped us achieve the new state-of-the-art performance.

3 | Model design

All the models proposed in the paper consist of two modules, the *Visual Features Extractor (VFE) Module* and the *Audio Feature Extractor (AFE) Module*.

3.1. Benchmark model

This model is the one we will compare the more complex models to, in order to determine the impact of the middle-level interactions between layers of the VFE and AFE modules. Figure 3.1 shows a diagram describing this model: one detail to take note of is the lack of interaction between the two modules up until the very last layer, before which the outputs of both modules are concatenated.

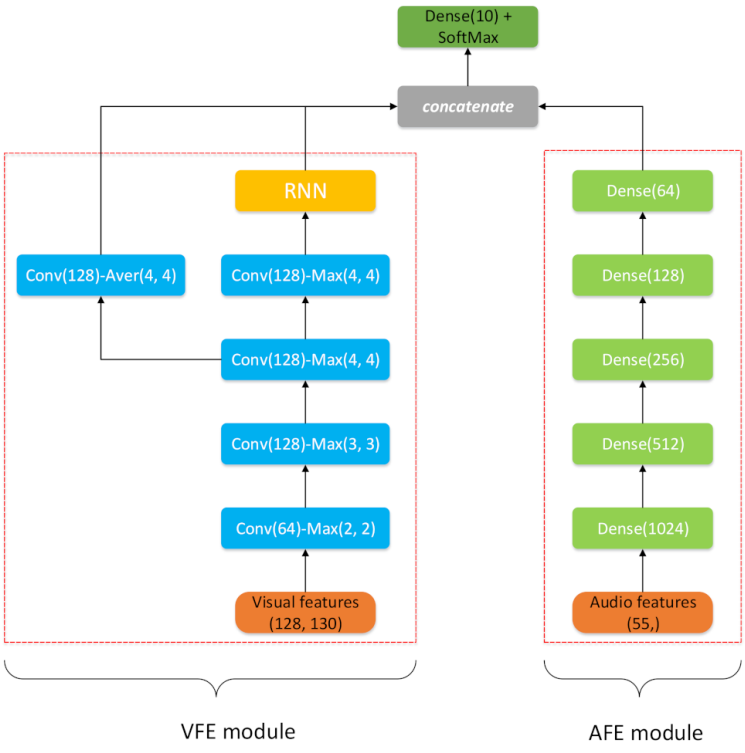

Figure 3.1: The original network architecture.

Each of the convolution blocks present in the VFE module can be further decomposed, as can be seen in Figure 3.2, into a Convolution layer, a Batch Normalization, a Max/Average Pooling and a Dropout layer.

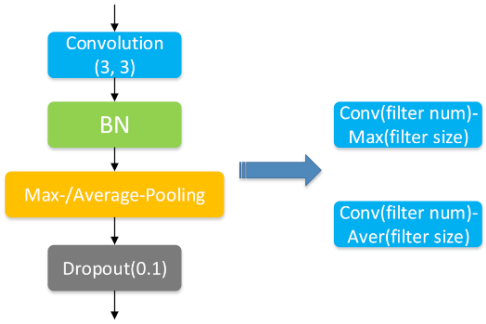

Figure 3.2: A single convolution block: each one can either implement max or average pooling.

3.2. Improved (MLFI) models

Three types of middle-level interactions are considered: one-way (audio), one-way (vision), and two-way interaction modes.

3.2.1. One-way (audio) interaction mode

As depicted in Figure 3.3, the visual features play a complementary role in this mode. Eight possible models have been explored, as a result of choosing either sub-mode A or sub-mode B, as well as only one of the four concatenation paths at a time.

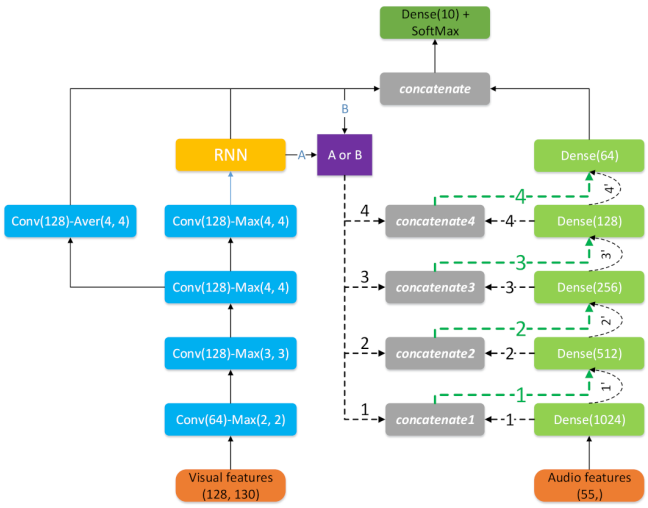

Figure 3.3: The network architecture using one-way (audio) interaction.

3.2.2. One-way (vision) interaction mode

In this case, the roles of the two modules are reversed: Figure 3.4 shows how five different models can be observed in this mode, depending on which AFE layer output is chosen to be concatenated to the **Conv(128)-Max(4,4)** block of the VFE module, before being fed to the 2-layer RNN.

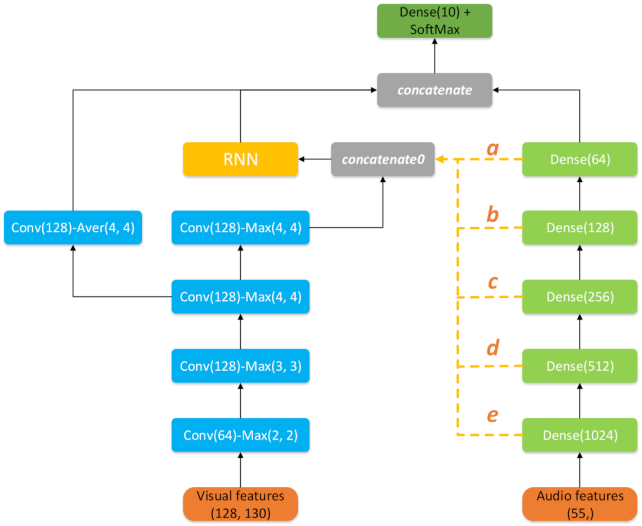

Figure 3.4: The network architecture using one-way (visual) interaction.

3.2.3. Two-way interaction mode

As one can imagine, this mode is a combination of the previous two: the AFE module provides four sets of middle-level features, and the corresponding paths b, c, d, and e; we can also distinguish between sub-mode A and sub-mode B, therefore creating eight more models to analyze.

4 | Training and Results

4.1. Training details

All files in the GTZAN dataset are transformed to Mel-frequency spectrograms and audio features separately by the preprocessing schema presented in Chapter 2: the Mel-frequency spectrogram with size 128 × 130 input to the VFE module and the audio features with the size 55 × 1 input to the AFE module.

For all other training details, we refer to the ones specified in the MLFI paper [1]: for instance, we trained each model using 10-fold cross-validation, splitting the dataset with a ratio of 7/2/1. As the hardware resources for the free use of Google Colab are limited, we were not able to train each fold for 150 epochs as mentioned in the paper: the average fold training ended around the 50th epoch.

We used the Adam optimizer with a starting learning rate of 0.01, using the *ReduceLROnPlateau()* callback as well, the cross-entropy loss function, and the classification accuracy as the observed metric. The choice of metric makes sense given the task and the fact that the samples in the dataset are evenly split among the 10 genres.

All the work was done on the Google Colaboratory platform, using Keras and Tensorflow to build our models, and the Librosa module to process the audio data.

4.2. Classification results on GTZAN

4.2.1. One-way interaction (audio) mode

As already mentioned in the paper, for both A and B sub-modes, the models with a middle-level interaction that is close to either the first or last layer of the AFE module seem to perform better. Analyzing the results, the highest accuracy for sub-mode A is 94.89%, and the average is 94.6%; as for sub-mode B, the highest is 95.06%, while the average is 94.66%.

While it seems as if sub-mode B performs a bit better, we are reluctant to claim as such: the difference between the two is quite small, and might also be a product of the stochasticity of the training, given that the number of epochs we trained each fold for might be too small, increasing the variance of the results.

Learning Feature (A or B)	Paths	Dense Layer Marker	Accuracy (%)
A	[1] 2' 3' 4'	1024	94.83
A	1' [2] 3' 4'	512	94.44
A	1' 2' [3] 4'	256	94.24
A	1' 2' 3' [4]	128	94.89
			Avg: 94.6
B	[1] 2' 3' 4'	1024	94.95
B	1' [2] 3' 4'	512	94.40
B	1' 2' [3] 4'	256	94.24
B	1' 2' 3' [4]	128	95.06
			Avg: 94.66

Table 4.1: The experimental results of one-way interaction (audio) mode. If the path label is marked with "[ ]", it means that the path implements the middle-level learning feature interaction method.

4.2.2. One-way interaction (vision) mode

In this case, the pattern observed in the one-way (audio) mode, according to which the accuracy of more "external" interaction paths is higher than for the others, does not seem to be present. This is also something that we can notice in the original paper's results.

Input Size of RNN	Paths	Dense Layer Marker	Accuracy (%)
(72,16)	e	1024	94.40
(40,16)	d	512	94.26
(24,16)	c	256	95.10
(16,16)	b	128	94.63
(16,12)	a	64	94.84
			Avg: 94.645

Table 4.2: The experimental results of one-way interaction (vision) mode.

4.2.3. Two-way interaction mode

With the combination of the two modes, we can see how the pattern observed for the one-way (audio) mode is missing. Despite that, the average accuracy is definitely higher than the other ones, suggesting that this double interaction is a good improvement to the first two proposed architectures.

The best model overall also belongs to this set: this is a different result compared to the one in the paper, which we attribute to the missing normalization of the input data on their part. However, the average accuracy is higher in the case described in [1], as well.

Learning Feature (A or B)	Paths	Dense Layer Marker	Accuracy (%)
A	[1] 2' 3' 4' e	1024	94.47
A	1' [2] 3' 4' d	512	94.74
A	1' 2' [3] 4' c	256	94.89
A	1' 2' 3' [4] b	128	95.00
			Avg: 94.77
B	[1] 2' 3' 4' e	1024	94.81
B	1' [2] 3' 4' d	512	**95.26**
B	1' 2' [3] 4' c	256	94.48
B	1' 2' 3' [4] b	128	94.76
			Avg: 94.83

Table 4.3: The experimental results of two-way interaction mode. The best model overall is highlighted.

4.3. Comparison of each mode

All the MLFI models perform, even if slightly in some cases, better than the original architecture: the benchmark model's performance is in fact 94.11%.

Figure 4.1 shows in a more intuitive way the above considerations.

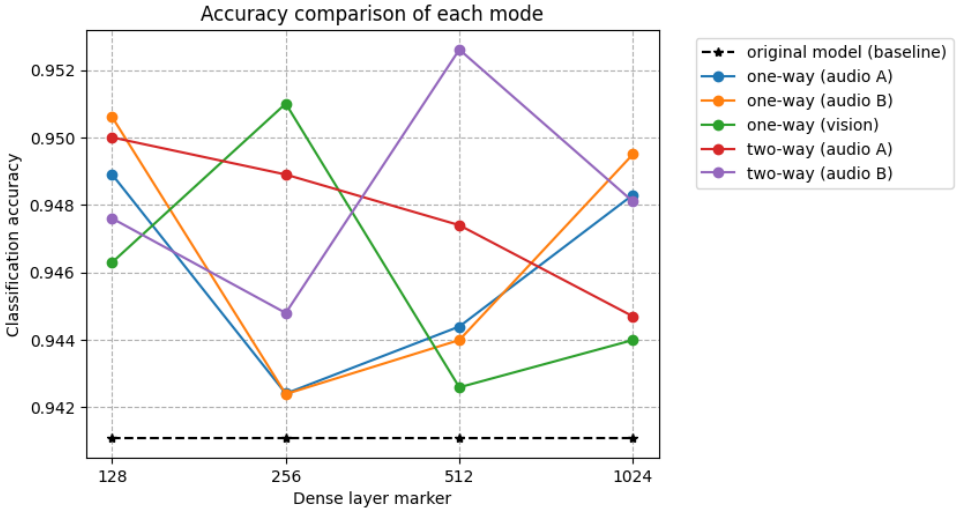

Figure 4.1: Accuracy comparison of each mode.

Since the models are evaluated based on classification accuracy, we analyze the confusion matrix of the two-way interaction (sub-mode B) model, our best-performing one, shown in Figure 4.2. One can notice a very good performance overall, with some drop in accuracy when it comes to country, hip-hop, jazz and rock music.

Although the numbers are a bit higher, the considerations are similar to those mentioned in the paper [1].

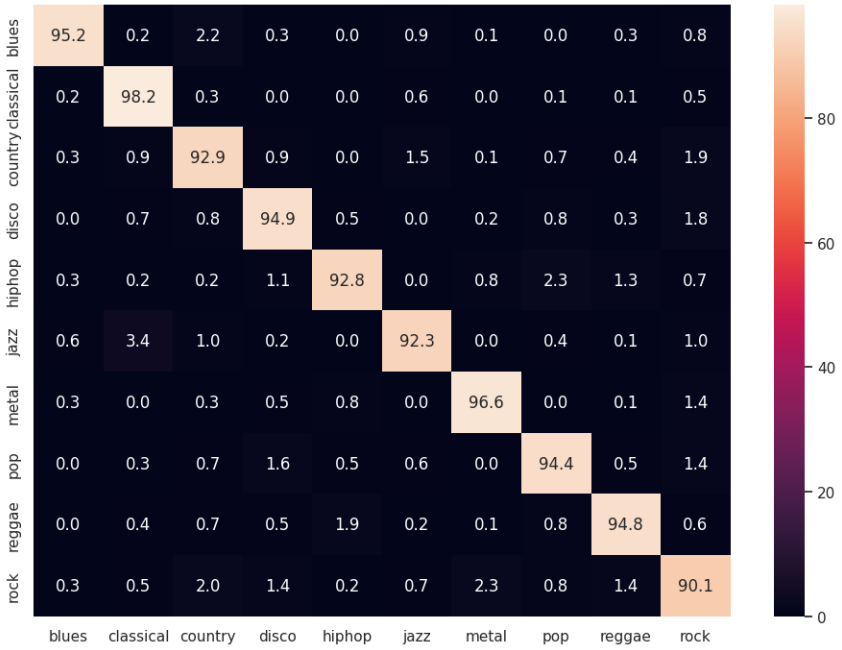

Figure 4.2: Optimal confusion matrix of two-way interaction (sub-mode B) mode.

5 | Conclusions

The paper [1] we re-implemented for this project proposed a new method (MLFI) to address the lack of interaction between branches of previously seen models for the task of music genre classification. They also showed the effectiveness of their method by comparing their best result, a 93.65% accuracy, to the previous state-of-the-art performance achieved by the BBNN model [2], with an accuracy score of 93.7%.

However, we found a critical issue in the first paper [1], that is to say the lack of data normalization in the pre-processing phase. By implementing this additional step, we allowed the models to better learn from the input data, which allowed us to achieve a new state-of-the-art performance, with our best-performing model reaching a 95.26% accuracy.

We are truly satisfied with the results of this project, both for the challenge of re-implementing a complex experiment, and for achieving a new state-of-the-art performance on the task of music genre classification.

Bibliography

[1] Jinliang Liu, Changhui Wang, and Lijuan Zha. A middle-level learning feature interaction method with deep learning for multi-feature music genre classification. *Electronics*, 10(18), 2021.

[2] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification. *Multimedia Tools and Applications*, 80(5):7313–7331, Feb 2021.