Music Genre Classification

Speech Lenguage Processing with Deep Learning - Universitat Politècnica de Catalunya

Gerard Gállego, Carles Garcia Cabrera, Anna Cuxart Garcia, Guillem Cortès

***Abstract*—Can you listen to a song and call its genre? How many seconds would you need? A song's genre is not something you can categorically classify with just one tag. You could tag a song with more than one genre and make sense anyway. That's because the definition of a genre is quite subjective. In this paper, we propose 2 methods in order to automatically classify a song's genre using Deep Learning.**

I. Introduction

Let's talk about music for a second. How often do you listen to music? And, what's more important, which platform do you use? Nowadays, both Spotify and YouTube with a net worth of some billion are reigning the music streaming market. They have millions of songs in their databases and claim to have the right music score for everyone. They are constantly suggesting us new songs to discover, as well as creating automatic play-lists based on the artists and also the genres. But, how can they identify it? Each genre has different music instruments, tone, rhythm, beats, flow etc. Isn't it quite a subjective feature? Music genre can be automatically classified using Artificial Intelligence, which actually saves both time and budget. In this paper, after a detailed research of the actual state of the art, we present two different music genre classification systems that have been implemented using Deep Learning. We discuss and compare them, regarding their complexity and the final results obtained.

We have created a ***Google Cloud*** server using the Google's Deep Learning version. It has allowed us to process the hardest computations thanks to the GPU that was offered.

We have been working with Git, in order to properly organize our implementations, hence you can find all our developed code in our ***Git*** repository: https://github.com/guillemcortes/slpdl-mgc.

II. State of the art

Since the first appearance of the Neural Networks, many fields have benefit from their awesomeness and performance to enhance predictions, classifications, even security. It's obvious and more than proven that Neural Networks can extract a lot of features from a signal, outperforming the traditional approaches. However, in some cases, there is a combination of handcrafted features and Neural Networks. Here we enumerate the latest techniques regarding the music genre classification topic.

In [1] they use a combination of MFCCs + CNN that gave them very poor results 17.18% accuracy , but they were able to enhance that result by changing the MFCCs with MEL spectrum 46.87% accuracy.

In [2] they use normalized MFCCs with 2 Dense Layers. They achieve 53% accuracy. Deepsound [3] presents a complicated combination of MEL-Spectogram CNN + LSTM + FC that rocket themselves to 67% accuracy.

In [4] they took inspiration from [3] to present 2 different approaches: one using 1D Convolutions from [5], LSTM and Dense Layers. And another one based on the Parallel CNN-RNN model from [6]. 5 2D convolutional blocks, Recurrent block (2D Maxpooling + Bi-directional GRU and dense layer at the end. They achieved an accuracies of 53% & 51% respectively.

This shows that it is far from being solved, and that there is still some work to do to achieve a very very good accuracy. On the other hand, though, it shows that it is more complicated problem than others because many times it is very difficult for us to classify a song in genres.

III. Datasets

In order to know which dataset could we use for our project, we firstly did an accurate research about the different possibilities, therefore we analyzed and discussed the following ones:

A. GTZAN Genre Collection [7][8]

This dataset was used for " Musical genre classification of audio signals " by G. Tzanetakis and P. Cook in IEEE Transactions on Audio and Speech Processing 2002 [9]. So, even though it is quite old, it is frequently used for many researchers in music classification field. It is composed by $1,000$ WAV audio tracks of 30 seconds, registered in Mono with a fs = $22050Hz$. They are classified in 10 different genres (100 tracks each): blues,classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. This is the first dataset that we used for our project, as it is quite common in this field, and we split it with a distribution of 60/100 train, 20/100 validation and 20/100 for testing. But unfortunately it was not very large, so we changed it afterwards in order to test our system with a bigger dataset.

B. FMA: A Dataset For Music Analysis [10] [11]

For all his 106,574 tracks, this dataset presents a collection of track's features, such as: ID, title, artist, genres, tags and play counts. It contains 163 genre, with their names and parent (used to infer the genre hierarchy and top-level genres). Also, it presents some common features extracted with librosa and audio features provided by Echonest (now Spotify) for a subset of 13,129 tracks. After testing our system with the GTZAN dataset, we wanted to experiment with a second dataset in

order to compare them and improve our results. The FMA presents 4 sub-datasets, regarding the number of tracks: Small (8,000 tracks of 30s, 8 balanced genres), Medium (25,000 tracks of 30s, 16 unbalanced genres), Large (106,574 tracks of 30s, 161 unbalanced genres) and Full(106,574 untrimmed tracks, 161 unbalanced genres). So we decided to use the small version and split each song in three as data augmentation. Therefore we had 24.000 tracks, which was more complex than the 1.000 tracks GTZAN dataset.

C. Music Acoustic benchmark [12]

It contains 1886 songs all being encoded in MP3 format. The frequency and bit-rate of these files are $44,100Hz$ and $128kb$ respectively. They are all classified non uniformly into 9 kinds of genres, which are: Alternative (145 tracks), Blues (120), Electronic (113), Folk/Country (222), Funk/Soul/R&B (47), Jazz (319), Pop (116), Rap/Hip-hop (300) and Rock (504).

D. Acoustic Brainz [13]

The Acoustic Brainz project aims to crowd source acoustic information for all music in the world and to make it available to the public. This acoustic information describes the acoustic characteristics of music and includes low-level spectral information such as key, chord keys, bpm, and beat count and high level information for genres, moods, scales, voice, gender, danceability, tonal, timbre, and many more. It provides 4 different genre propabilities: Tzanetakis model, Electronic classification, Dortmund model and Rosamerica mode.

IV. General Setup

Our main objective is to design and implement an algorithm to automatic classify a songs genre (e.g. Rock, Pop, Jazz, ...) from its audio signal and from some features that can be extracted. For this project, in order to achieve this goal, we are going to work with Deep Learning algorithms. We will start by extracting a simplified representation of each song from our data set and train a deep neural network to classify the songs for their genre. Then, we will use it to classify new songs with an unknown genre.

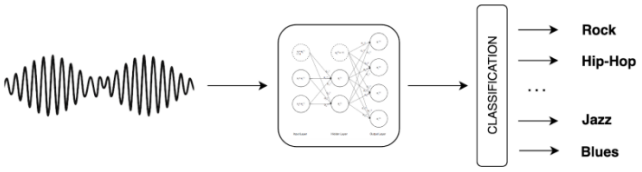


Fig. 1: General Setup

V. Implementation

After researching about the actual state of the art, we decided to design and implement two different structures for our genre classification goal. This would allow us to gain more experience and knowledge about these deep learning algorithms and compare their results to discuss which one was more accurate. For both implementations, we have used the Adam Optimizer and the Cross Entropy Loss criteria, as well as ***Pytorch*** and ***Librosa*** packages.

Firstly, in order to have a first approach, we used the GTZAN dataset, that's composed by $1,000$ tracks that are classified in 10 different genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. With it, we created our own library using Librosa for numerical feature and spectrogram extraction, in order to easily import it from our notebook. Thanks to this methodology, we realized that processing all 1000 audios every time was quite heavy, so we decided to do it once and save the extraction as ***pickles***, instead of repeating this process each time.

Afterwards, we decided to test it with a larget dataset, and for that we choosed the small version of FMA which was composed by $8,000$ tracks of 30 seconds, that were classified into 8 balanced genres: experimental, electronic, rock, instrumental, pop, folk, punk and avant-grade. Once we started to explore this second dataset, we realized that the 8GB of tracks were in format MP3, so we had to convert them to .WAV, with raise it to $40GB$. In order to do that conversion, we implemented a ***Bash*** script to iterate all MP3 files. As we were working with ***Google Cloud***, we were able to process this data comfortably, instead of having to work with it locally. The genre's information in this dataset, as well as the tracks, were mixed in different CSV files with all the information from the full version, so we had to develop a specific script in order to extract them all.

A. Handcrafted features with a DNN

For the first implementation, we were inspired by two different proposals: Identifying the Genre of a Song with Neural Networks from N.Singh and Music Genre Classification with Python. A Guide to analysing Audio/Music signals in Python. from P.Pandey. [14] [15] respectively. They were both working with feature extraction applied to a DNN. The first project was using 20 Mel Frequency Ceptral Coeficients from each song to train the net while the second one was working with the following features: Chroma Short Time Fourier Transform, Spectral Centroid, Spectral Bandwidth, Spectral Rolloff and the Zero-Crossing Rate among the others. So we decided to combine their ideas and train a DNN with all the features and the MFCC, which we have firstly standardized. We have distributed the data as 80% training, 10% test and 10% for validation.

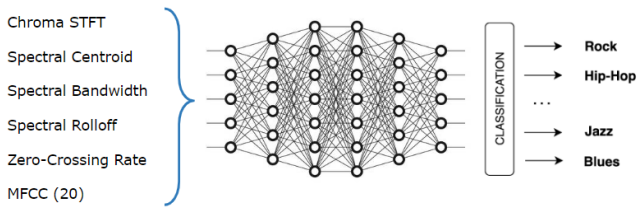


Fig. 2: First implementation structure: Handcrafted Features + DNN

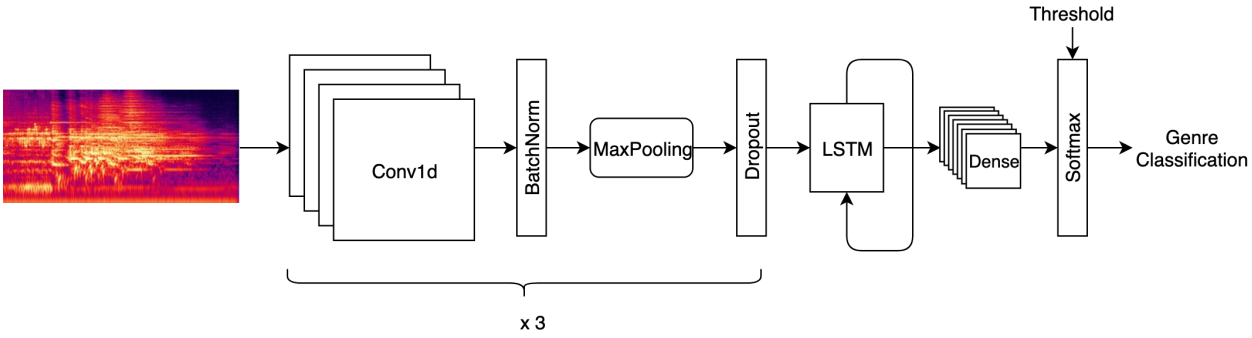


Fig. 3: Third implementation structure: CRNN (MEL spectrogram + CNN-1D + LSTM)

Our implemented system consists in 5 fully-connected layers with ReLU activation and a dropout in the hidden layers. We also added batch normalization.

B. MEL spectogram with a 2-D CNN

As a second implementation, we decided to follow the proposal of M.Lachmish and his "Music Genre Classification" project from the Tel Aviv University [16]. He suggested to work with a two dimension Convolutional Neural Network, trained with the MEL spectrogram of each song's audio signal as an image. The network consists of 3 convolutional layers, with a max pooling between the layers and dropout. Then, we flattened the output to fed a fully connected layer activated with a softmax function in order to classify the song in one of the 10 different genres.

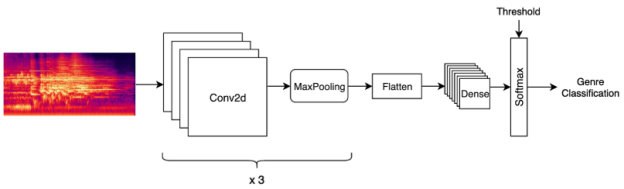


Fig. 4: Second implementation structure: Mel Spectrogram + CNN-2D

As it can be easily seen in 4 our system consists in 3 convolutional layers with maxpooling and batch normalization and a Dense layer at the end. There is also dropout to prevent overfitting.

C. CRNN

As a third and last approach (Fig. 3), inspired by [5], we decided to go for a more complex approach. It consists in a 3 convolutional layers of 1D convolutions with its max pooling, batch normalization and dropout. And after that, a 2-Layer LSTM. The assumption underlying this model is that the temporal pattern can be aggregated better with RNNs than CNNs, while relying on CNNs on input side for local feature extraction. In CRNN, RNNs are used to aggregate the temporal patterns instead of, for instance, averaging the results of the convolutions and sub-sampling as in other CNNs.

VI. Results

Here we summarize all the results we obtained and we explain the implementation process.

As mentioned before, we developed three different approaches using the GTZAN dataset and, additionally, we did an hyperparameter optimization for the DNN system.

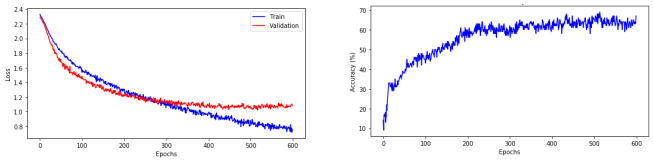
Finally, we tested the optimized DNN model with the FMA dataset, which is wider than the GTZAN.

A. Handcrafted features with a DNN

1) Early Results: We used these hyperparameters:

- Hidden sizes = [256, 128, 64, 32]
- Learning rate = 0.0001
- Dropout = 0.5
- Number of epochs = 600

We obtained an Accuracy of **64%** on the test set.



Loss Accuracy

Fig. 5: [GTZAN] Losses and accuracy for the Hand Crafted Features + DNN implementation

2) After hyperparameters fine tunning: We performed a grid-search to find optimal hyperparameters, during 200 epochs:

- Hidden sizes:
  [256, 128, 64, 32]
  [512, 256, 128, 64]
  [128, 64, 32, 16]
- Learning rate: 0.001, **0.0001**, 0.00005
- Dropout: 0.1, **0.25**, 0.5

After hyperparameter optimization we obtained an Accuracy of **67%** on the test set.

3) Using FMA Dataset: We tested this system with the FMA Dataset instead of GTZAN, using the optimized hyperparameters. The Accuracy drops to **44%** on test set.

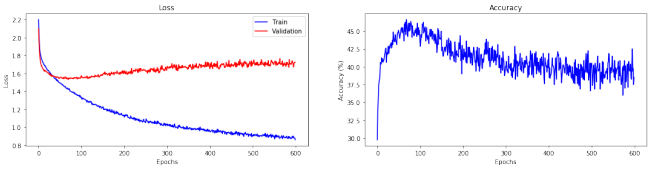


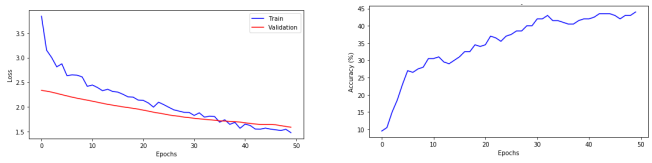
Fig. 6: [FMA] Losses and accuracy for the Handcrafted Features + DNN implementation

B. MEL spectogram with a 2-D CNN

We used these hyperparameters:

- Num channels = [1, 32, 64, 128]
- Kernel Size = 5x5
- Padding = 2 (full padding)
- Learning rate = 0.0001
- Dropout = 0.5
- Number of epochs = 50

We obtained an Accuracy of **56%** on the test set.



Loss Accuracy

Fig. 7: [GTZAN] Losses and accuracy for the MEL spectrogram with the 2D CNN implementation

C. CRNN

We used these hyperparameters:

- Num channels = [128, 64, 32, 16]
- Kernels size = [5, 3, 3]
- Padding = [2, 1, 1] (full padding)
- Learning rate = 0.0001
- Dropout = 0.3
- Number of epochs = 500

We obtained an Accuracy of **53%** on the test set.

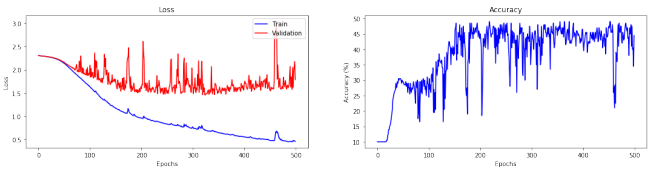


Fig. 8: [GTZAN] Losses and accuracy for the CRNN implementation

VII. Conclusions

We have mainly worked with two different kind of approaches, one using features as inputs and another using spectrograms. That has lead us also to play with 3 different kind of neural networks, a dense network for the features and two different convolutionals networks for the spectrograms, one of them attached to a recursive one.

If we compare the DNN implementation with the 2D CNN, we can observe that we achieved a higher accuracy (64%) in the first implementation than in the second one (56%), even though we can appreciate that there is overfitting in the deep neural network.

Regarding the two MEL Spectrogram implementations, we have achieved a better approach with the 2D CNN than the CRNN, with an accuracy of 56% and 53% respectively, although the second one is an upgraded and more complex version. Furthermore, if we take a closer look at the CRNN Loss and Accuracy functions, we can appreciate that there is an uncommon behaviour, so it should be improved with a readjustment of the hyperparameters. If we compare it with the CRNN of the state of the art, we can observe that it was actually their best approach, with an accuracy of 67%, while we reach the same accuracy in our best implementation, the Handcrafted features with a DNN.

Finally, even though the FMA dataset is 8 times larger than the GTZAN and we used its optimized hyperparameters, we have obtained better results using the GTZAN in the DNN approach, with an accuracy of 67% versus the 44% of FMA. Also, if we pay attention to its accuracy, it can be observed that there's overfitting.

During the project development we have seen our technical skills improved, we have been using new libraries and tools that helped us while pursuing better results. Also we have seen improved our abilities to work as a team, helped by tools that we have discovered meanwhile and by learning from each other.

Together we have learned to share our knowledge and help each other, facing the challenges that this project has brought and overcoming them as a team. We have learned to work and properly structure our project with Git and to work with Pytorch and Librosa libraries to develop our implementations.

We have overcame different peculiarities that we have met on our way, such as exploring our input data and designing specific algorithms to pre-process it. We have also dealt with new network architectures such as 1D convolutions or the LSTM.

References

[1] Matan Lachmish. *Music Genre Classification. Classify music genre from a 10 second sound stream using a Neural Network.* https://github.com/mlachmish/MusicGenreClassification. [Online; accessed 06-June-2019]. 2018.

[2] Navdeep Singh. *Identifying the Genre of a Song with Neural Networks*. https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners/blob/master/Chapter04/GenreIdentifier.ipynb. [Online; accessed 06-June-2019]. 2018.

[3] Piotr Kozakowski & Bartosz Michalak. *Music Genre Recognition.* http://deepsound.io/music_genre_recognition.html. [Online; accessed 06-June-2019]. 2016.

[4] Priya Dwivedi. *Using CNNs and RNNs for Music Genre Recognition.* https://towardsdatascience.com/using-cnns-and-rnns-for-music-genre-recognition-2435fb2ed6af. [Online; accessed 06-June-2019]. 2018.

[5] Keunwoo Choi et al. “Convolutional recurrent neural networks for music classification”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 2392–2396.

[6] Lin Feng, Shenlan Liu, and Jianing Yao. “Music genre classification with paralleling recurrent convolutional neural network”. In: *arXiv preprint arXiv:1712.08370* (2017).

[7] Bob L Sturm. “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use”. In: *arXiv preprint arXiv:1306.1461* (2013).

[8] Jakob Leben. *Music analysis, retrieval and synthesis for audio signal.* http : / / marsyas . info / downloads / datasets.html/. [Online; accessed 06-June-2019]. 2015.

[9] G. Tzanetakis and P. Cook. “Musical genre classification of audio signals”. In: *ieee* (2016). URL: https : // ieeexplore.ieee.org/document/1021072.

[10] Michaël Defferrard et al. “Fma: A dataset for music analysis”. In: *arXiv preprint arXiv:1612.01840* (2016).

[11] Michaël Defferrard et al. “FMA: A Dataset for Music Analysis”. In: *18th International Society for Music Information Retrieval Conference*. 2017. URL: https : //arxiv.org/abs/1612.01840.

[12] Technische Universitaet Dortmund. *Acoustic Breinz.* http://www-ai.cs.tu-dortmund.de/audio.html. [Online; accessed 06-June-2019].

[13] *Acoustic Brainz.* https : // acousticbrainz . org/. [Online; accessed 06-June-2019].

[14] Navdeep Singh. *Identifying the Genre of a Song with Neural Networks.* https://medium.com/@navdeepsingh_2336/identifying-the-genre-of-a-song-with-neural-networks-851db89c42f0. [Online; accessed 06-June-2019]. 2018.

[15] Parul Pandey. *Music Genre Classification with Python. A Guide to analysing Audio/Music signals in Python.* https : / / towardsdatascience . com / music - genre - classification - with - python - c714d032f0d8. [Online; accessed 06-June-2019]. 2018.

[16] Matan Lachmish. *Music Genre Classification.* https : / / medium . com / @matanlachmish / music - genre - classification - 470aaac9833d. [Online; accessed 06-June-2019]. 2018.