Multimedia Tools and Applications (2023) 82:3911–3930
https://doi.org/10.1007/s11042-022-12523-w

CONEqNet: convolutional music equalizer network

Jesús Iriz[1] · Miguel A. Patricio[2] · Antonio Berlanga[2] · José M. Molina[2]

Received: 5 May 2021 / Revised: 18 November 2021 / Accepted: 25 January 2022 /
Published online: 18 July 2022

Abstract
The process of parametric equalization of musical pieces seeks to highlight their qualities by cutting and/or stimulating certain frequencies. In this work, we present a neural model capable of equalizing a song according to the musical genre that is being played at a given moment. It is normal that (1) the equalization should adapt throughout the song and not always be the same for the whole song; and (2) songs do not always belong to a specific musical genre and may contain touches of different musical genres. The neural model designed in this work, called CONEqNet (convolutional music equalizer network), takes these aspects into account and proposes a neural model capable of adapting to the different changes that occur throughout a song and with the possibility of mixing nuances of different musical genres. For the training of this model, the well-known GTzan dataset, which provides 1,000 fragments of songs of 30 seconds each, divided into 10 genres, was used. The paper will show proofs of concept of the performance of the neural model.

Keywords Music equalization · Convolutional neural network · Music information retrieval

1 Introduction

It is a fact that music has a social meaning. In the field of psychology, it has been shown that music has several functions. Some of these functions relate to social combination in our everyday lives and the control of emotions [29]. Music functions socially by managing

✉ Miguel A. Patricio
mpatrici@inf.uc3m.es

Jesús Iriz
jesus.iriz.gonzalez@gmail.com

Antonio Berlanga
aberlan@ia.uc3m.es

José M. Molina
molina@ia.uc3m.es

[1] MasMovil Engieneering Team, Madrid, Spain

[2] Applied Artificial Intelligence Group, Universidad Carlos III de Madrid, Colmenarejo, Madrid, Spain

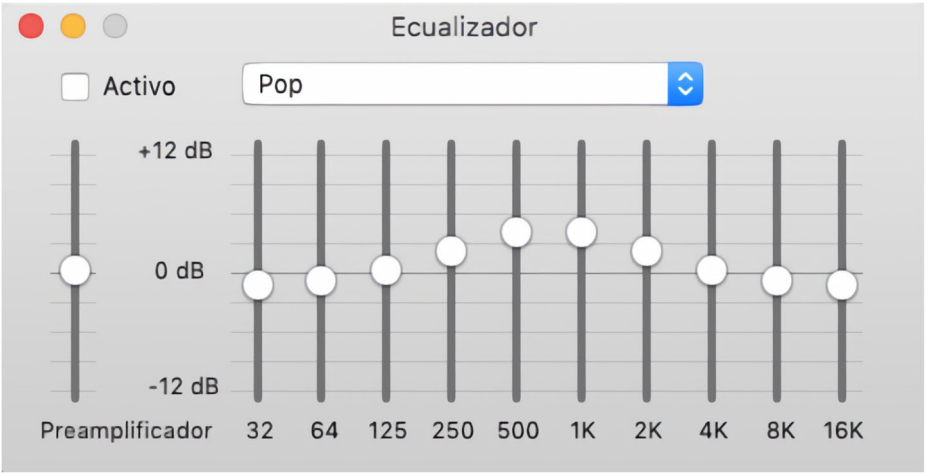

Fig. 1 EQ profile for iTunes [1] Pop

ourselves, interpersonal relationships and the environment around us [13]. In this way, music plays an increasingly important role in everyday life.

Music is sound, sound is vibration, and vibration is energy that is transmitted in the form of waves that reach the ear and from there travel to the brain. These vibrations transmit a message that can be more or less significant, awakening feelings that are pleasant, exciting, calming, etc. Each musical instrument, like any other sound source, produces sound in a certain area of the audible frequency spectrum. Some cover more space and others less. This is where equalizers come in; equalizers are devices that allow us to alter the frequency response of a sound by boosting or attenuating certain frequencies. Equalizers can be used as filters to attenuate or remove annoying frequencies, noise or interference that mix with the sound. They can also be used to vary the character of an instrument. There are different types of equalization [38]. In our specific case, we will focus on parametric equalization [4]. We understand equalization as the process of altering the amplitude of each of the frequencies of audio to make some frequency ranges more noticeable and blur others.

The ISO standard states that the frequency bands must be at least 32, 64, 125, 250, 500, 1000, 2,000, 4,000, 8,000 and 16,000 Hertz. A volume variation between −12 decibels and +12 decibels is applied to each of these bands. A good equalization, even if it is slightly subjective, seeks to amplify the most common frequency ranges of the song so that the most important details stand out more over the less important ones.

In Fig. 1, we can see the EQ profile for the Pop music genre. This profile seeks to boost frequencies close to 500 Hz and 1000 Hz. This is because the predominant sounds in pop music, such as vocals, guitars and synthesizers, are around those frequencies. The usual female pop voices hover between A3 and C6 at frequencies of 220 Hz and 1,046 Hz, respectively, so it is logical that these ranges are amplified. However, very high and very low sounds are not usually used in this musical genre since they do not match the frequency range of any usual pop instrument.

If an instrument can be broken down into simple frequencies, a piece of music with multiple instruments can also be broken down into a set of simple frequencies that this time can already be understood by a computer. Musical genres are groups of sonorities and techniques that give as a product a recognizable and distinguishable ensemble due to their

characteristics of other genres. The division of music by musical genres can be as exhaustive as desired; two genres can be distinguished, for example, instrumental music that includes instruments and vocal music that, in addition to instruments, includes voice. Even if we want to, we can differentiate genres according to the peculiarities they have according to their origin, separating between western pop and eastern pop, if we want.

Musical genres can also be understood as a tree diagram in which there is a hierarchy depending on whether some genres are derived from others [11]. For example, jazz is a genre derived from a mix of rock and roll and African-American music.

Therefore, we can say that music has a hierarchical structure, which can be divided into three levels of semantic representation: low, middle and upper. The lower level is composed of timbral features usually characterized by the spectrum derived from a short-time sliding window. For the characterization of these timbral features, one can use the spectral centroid/roll-off/flux, Mel-frequency cepstral coefficients (MFCC), octave-based spectral contrast, and time-varying features such as zero-crossing and low energy. These low-level features combine with each other to form other mid-level features, obtaining descriptions of rhythm (beat and tempo) and pitch content and giving an idea of the regularity of the music. Finally, there is the upper level, which combines mid-level features to define what is meant by genre. This hierarchical structure allows us to make use of tools such as convolutional neural networks (CNNs) that have been so successfully used for image classification [17, 31, 39], natural language processing [6, 25], and other fields [2, 8, 21, 30]. For instance, in the field of image classification, the first layers of CNNs extract the basic characteristics of the images, which are combined in successive layers to obtain a more symbolic representation of the image, to arrive at the final result of semantic labelling of the image.

Digital music players usually have features to establish the desired equalization pattern. Likewise, there are pre-established patterns, as indicated in Fig. 1, for different musical styles. Although these patterns allow an optimal equalization for a musical style, each song has its optimal equalization, which is different from the rest of the songs. Our model proposes the most useful equalization for each genre, even if they are not perfect for each song. In addition, equalization patterns are set for the entire song, regardless of the different variations it may suffer. This work shows an adaptive system that identifies, for a certain segment of the song, the most likely musical styles.

Our work will focus on the design of a new neural model called CONEqNet (convolutional music equalizer network). This model will make use of the hierarchical structure of music to determine the mix of genres of a given musical piece at a given time. Once this information has been obtained, the predetermined profiles of these musical genres are integrated in a weighted way, obtaining a unique profile for a given song at a given time. In this way, we can have an intelligent, adaptive and song-specific equalization. The present work is organized as follows. In the first section, the current state of the art is presented, mainly in relation to the classification of musical genres of songs. Next, the proposal of a neuronal model that allows us to obtain a sequence of equalization profiles for a given song is presented. In the following section, proofs of concept of the design of the new neural model CONEqNet are carried out, ending with a section that presents the conclusions of this study.

2 Related works

Music information retrieval (MIR) problems consist of classifying musical content to retrieve information about the instruments played, the type of music, and the composer or

genre to which it belongs [32]. The classification of a musical genre is not a trivial task; it is a challenge because it involves a large number of variations in musical instruments, harmonies, and melodies or timbres. Even though this is a topic that we discuss in our work, it is normal that in one piece of music, there can be two or more musical genres. In our work, we present a neural model that allows us to recognize at a given moment the different musical genres that are playing to automatically choose the best-suited equalization profile. The classification of musical genres has been treated by various supervised classification methods [24, 26]. One of the first works was that developed in [37], where K-nearest neighbor (KNN) was used. Other works have been based on other computational models, such as Gaussian mixture models (GMMs) [15, 16], hidden Markov models [19], linear discriminant analysis [5], support vector machines [10, 23], artificial neural networks [22], and convolutional neural networks [7, 20, 28].

For this, they proposed segmenting the songs and using all the fragments for the process. Other studies use only the main part of a song to classify it, but this implies a significant loss of information from the rest of the song. Other advantages offered by their segmentation proposal are that, in training, more data are available and that in the testing phase, the results will be more accurate by being able to calculate an average and determine a correct classification for the total song and not just for a fragment. The neural network model used is a recurrent convolutional neural network (RCNN). This dataset is a homemade dataset consisting of 30 songs from ten different genres that correspond to the ten genres of GTzan [33].

In [12], also with the aim of classifying music by genres, the authors use a dataset of 400 songs of two genres, Indian music and classical music, with 200 songs in each of them. The data extracted from each song are the MFCCs (Mel Frequency Cepstral Coefficients) of the entire song. The contribution of this study is that they implement an intermediate layer that learns to transform the MFCCs into another series of song parameters, including beats per minute (BPM), the amount of voice, the number of instruments, and even a very curious parameter that they call 'Dance', which they define as the ability of a song to be danced to.

The neural network model that they use is a multilayer perceptron. The results obtained are not very precise (87% identify a song of classical music, and 82% one of Indian music), especially considering that it only distinguishes between 2 genres, which should be a much simpler task than with more categories. This study serves as a sign that there are more effective methods than a simple multilayer neural network and transforms the MFCCs into different parameters, such as those mentioned. In this way, and compared with the previous study, it is easy to conclude that using MFCCs as input parameters of the network is the best decision.

The authors in [9] use a convolutional neural network with the aim of analyzing whether it is possible to use a neural network to create playlists of recommendations on platforms such as Spotify based on the music they usually listen to. The idea is to train a network with the most listened songs of a user, observe their operation, and consider how to offer lists of similar recommendations. Again, the MFCCs are used as input parameters, but this time, they are extracted from just a fragment of each song. The output consists of a genre classification. An interesting aspect of this study is the analysis of what the network is learning. To verify it, it analyzes the outputs of the neurons of each layer individually, introduces a series of songs, and generates playlists in common that activate a specific neuron of that layer. In this way, it observes how as you progress through the layers, the deeper layers highlight more specific characteristics, such as genres, and the initial layers find patterns in the sound

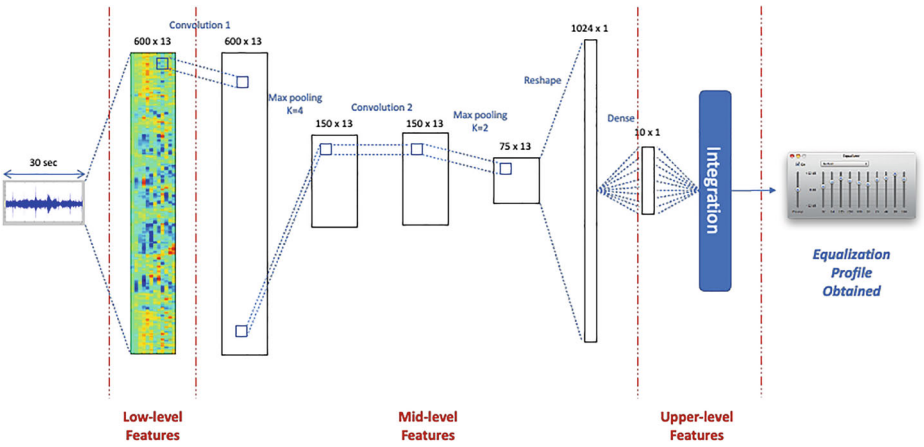

Fig. 2 Architecture of the smart equalizer proposed in this study. This architecture uses the hierarchical structure of music to determine the musical genres within a 30-second piece of music. As an output, the optimized equalization profile for that time is obtained

of the song. For instance, in the first layer, they discover that neurons 14, 250, and 253 identify vibrato, vocal thirds (voices singing over each other at a specific distance of 3 tones) and serious percussion. Thus, in the next layer, it identifies complete tunings of the song and even chords. Finally, in the layer before the output layer, it manages to identify some prototypes of truly specific final genres, such as Gospel, Christian Rock, Chiptune or Chinese Pop. This model identifies the techniques and sounds present in it, and by combining it, it gradually ends up obtaining a genre classification similar to the approach that a person would take to identify instruments, identify the way they sing, and end up identifying the genre.

The proposal of our work is to find a neural model that is able to determine the optimal equalization of a song at any given moment. The new CONEqNet model implements a lightweight convolutional network (it is designed to be run on devices with low hardware resources such as smartphones) to detect the musical genres that are playing and thus propose the best equalization at any given moment.

3 Proposed architecture

This section shows the architecture of the new neural model CONEqNet. In Fig. 2, this architecture is depicted. This architecture follows the hierarchical nature of the musical pieces: low-level, mid-level and upper-level.

3.1 Low-level features

The low level is composed of timbral features characterized by the spectrum derived from a short time sliding window. For the characterization of these timbral characteristics, we used the Mel Frequency Cepstral Coefficients (MFCC) [14]. The MFCCs are coefficients representing a sound wave derived from the coefficients of the scale of Mel. The Mel scale is a transformation applied to frequencies, transforming it from a linear scale to a perceptual scale. For instance, a human being does not perceive the difference between $100Hz$ and

$200Hz$ to be the same as that between $2,000Hz$ and $2,100Hz$, although the numerical distance between the two pairs is the same. With this transformation, on the one hand, the values introduced to the neural network are more "natural", which is more representative of how a person perceives them. On the other hand, it is possible to eliminate the number of frequencies necessary to obtain the same precision, since higher frequencies provide less information to the human ear. The conversion formula from Hertzs (**h**) to Mels (**m**) is given by:

$$\mathbf{m} = 1127.01048 \ln (1 + \frac{\mathbf{h}}{700}) \tag{1}$$

To obtain the MFCCs, the process is depicted in Algorithm 1.

Algorithm 1 Mel frequency cepstral coefficients.

Segment the sound into fixed length sections S_i;
for *each section* S_i **do**
  Apply the discrete Fourier transform to separate it into frequencies and obtain the spectral power (or relative energy) of each frequency in segment ;
  Apply the Mel scale (1) to the spectra obtained in the previous point;
  Take the Neperian logarithm of each Mel coefficient obtained;
  and Apply the discrete cosine transform to each of these logarithms;
end

Thus, we obtain temporal information for each song with each segment and frequency information with each coefficient obtained in each segment. In this way, although handling more data, the temporal information is blurred less than simply using Mel coefficients. As the result of calculating the MFCCs is a two-dimensional matrix, this makes them suitable for working with convolutional neural networks.

According to studies such as [36], the optimal number of frequencies to be taken to calculate the MFCCs is between 10 and 20. In the case of our work, 13 will be used following the indications of studies such as [9]. The consensus on how many windows to take in each fragment is about one window for every $0.05s$ of audio. In this way, if we use samples of $30s$ duration, we will have 600 window shots per song. The neural network will take as input a total of 600×13 parameters.

As we will see later, our work is based on the GTzan dataset [33] that provides 1,000 fragments of songs of 30 seconds each, divided into 10 genres: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae and Rock.

3.2 Mid-level features

Low-level features combine with each other to form other mid-level features that obtain descriptions of rhythm (beat and tempo) or pitch content. These features will be extracted by applying a convolutional neural network (CNN) architecture. In [9] a CNN with 3 hidden layers is used. In our proposal, we want to have a simpler neural model that can be embedded in platforms with fewer resources (for instance, smart speakers or smartphones). Although we are aware that precision is lost, as will be seen later, great precision is not needed when mixing equalizations of musical genres. Therefore, we proceed to make a simplification using only 2 hidden layers.

In this way, each of these layers will apply max pooling and dropout. Max pooling consists of applying a reduction of parameters so that as enters the network, fewer data and

networks begin to take shape in a given output (for example, rated 7,800 parameters in just 10 obtained genres). On the other hand, the dropout consists of eliminating a certain percentage of connections between layers in each iteration so that the network begins to adapt not only to its function of classifying but also so that each layer begins to learn to solve errors of the previous layers if they exist. Care must be exercised when handling both data you have to be careful, since a very extreme value could lead to poor results of the neural network.

Finally, just before the final output, a dense (or fully connected) layer will be used in which no max pooling will be applied; that is, it will maintain all connections with the previous layer.

Finally, this layer applies a rectified linear unit (ReLU) function, which is equal to 0 when its input is less than or equal to 0 and is linear when the input is positive.

3.3 Upper-level features

Finally, there is the top level. By combining mid-level features, a symbolic representation of the musical genres found in a piece of music at a given fragment of time is obtained. An equalization profile is defined as a certain attenuation or boost value (between -12 and 12 decibels) for sounds at frequencies 32, 64, 125, 250, 500, 1000, 2,000, 4,000, 8,000 and 16,000 Hertz. It can be defined as an array of 10 values between -12 and 12. Let E be the equalization profile obtained from the integration, e_i be the equalization profile for a given genre i (in our case, we have up to 10 different genres), and g_i be the probability that the musical segment belongs to genre i. At this point, we can process the output with a softmax function and simplify it to the most probable genre (2).

$$E = \{e_i \mid max\left(Softmax(g_i)\right),\ 1 \leq i \leq 10\} \tag{2}$$

Another possibility is to use the raw output as a probabilistic function and consider a set of the three most probable genres (3).

$$E\left[j\right] = \sum_{k=1}^{3} \frac{e_k[j] \cdot g_k}{3}\ ,\ \ j \in [32, 64, 125, 250, 500, 1000, 2000, 4,000, 8,000, 16,000]$$

$$[g_k, g_{k'}, g_{k''}] = Max_3(g_i), 1 \leq i \leq 10 \tag{3}$$

In this study, we use a total of three for experimentation. With this information, an integration of the predefined equalization profiles in each genre is carried out. In the softmax case, we will simply use the given genre's profile. For the probabilistic case, we will use a weighted approximation for the final profile as a mix of the three genre profiles. In this way, we obtain as output the optimal and specific equalization profile for this piece of music at that precise moment. As we will see in the experimentation section, this process is not carried out with the entire song but is applied to segments of the song, producing an equalization more adapted to the music that is heard in each moment.

3.4 Order of complexity of the architecture

In this section, we will analyze the order of complexity of each of the processes involved in the extraction of features from each of the levels of the proposed architecture. Both medium- and upper-level features are obtained with linear complexity ($O(N)$). The low-level features are obtained by calculating the mel frequency cepstral coefficients (MFCC). MFCC uses the

calculation of frequency domain features as one of its main processes. It also performs other calculations; however, these are of lower complexity than the fast Fourier transform (FFT). Press et al. [27] argue that the FFT can be calculated in $O(Nlog_2N)$ complexity. Therefore, obtaining the MFCC involves the same complexity, and accordingly, the complexity of the architecture can be expressed as $O(Nlog_2N)$.

4 Experimentation

4.1 Datasets

To design the CONEqNet network, the present work has been based on the GTzan dataset [33].

GTzan is the most commonly used dataset for works related to the classification of music by genres. Originally, it is the dataset that George Tzanetakis and Perry Cook created in their work "Music genre classification of audio signals", one of the pioneering works of classification of music in genres, so it is of special interest. The dataset consists of 1,000 fragments of songs of 30 seconds each, divided into 10 genres: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae and Rock. In addition, the fragments were collected from different sources, such as radio or music CDs, causing variation in the recording conditions of each.

Our goal is to detect the genre of music that is being listened to in order to carry out the optimal equalization of the fragment that is being listened to. Reviewing the equalization library offered by iTunes [1], we found that adjustments are available for 6 of the 10 GTzan genres, with a missing profile for Country, Reggae, and Disco. Therefore, the three genres have to be analyzed in instrumentation; based on that, new profiles must be designed, using as a reference a table of conversion of notes to frequencies and a table of frequency ranges for each instrument of the reference.

4.1.1 Country

The country is a genre that, although a cultural derivative of Rock & Roll, has an instrumentation very similar to Pop, so we will take the latter as a starting point. Some of the most notable instruments of this genre that are not usually present in Pop are the following:

- Violin: Its frequency range is from G3 to G7, which corresponds to frequencies of 196 Hz to 3136 Hz.
- Banjo: Its most common tuning corresponds to frequencies of 110 Hz to 800 Hz.
- Harmonica: Frequencies between 180Hz and 3100Hz.

In this way, based on the above and the equalization profile of Pop, we will slightly enhance the bands of 125 Hz, 1 kHz, 2 kHz and 4 kHz and somewhat enhance the bands of 250 Hz and 500 Hz.

4.1.2 Reggae

Reggae is a genre originally developed in Jamaica and derived from genres such as Rock or Ska. Reggae and Rock have in common that they both maintain serious instruments, such as electric bass and percussion; however, the vocals and guitars that are used in Reggae tend to be slightly sharper than in Rock. To shift the frequencies of these timbres slightly toward

higher tones, we blur the lower frequencies of these and enhance the higher frequencies. As a result, from the equalization profile of Rock, we will keep the frequencies up to 64 Hz, lower the frequencies of 125 Hz and 250 Hz, and raise those of 250 Hz and 500 Hz.

4.1.3 Disco

Disco, as the name suggests, is a genre that became popular in party halls (discos) from the seventies. Due to the sound limitations of the playback devices of the time, it was difficult to emit very high-pitched sounds on high-volume devices, so very high frequencies are not very prevalent in this genre. On the other hand, as a derivative of Pop and Blues, it maintains some of the bass sounds of blues with percussion and bass and the central and not very sharp sounds of Pop by adding instruments such as synthesizers. Therefore, we will start from the Blues and increase the frequencies of 250Hz and 500Hz, lowering the frequencies of 4kHz, 8kHz, and 16kHz.

4.2 MSD (Million Song Dataset)

The Million Song Dataset (MSD) [3] is a very broad dataset that has a total of one million songs. This dataset has been used to validate the neural model designed and trained with the GTzan dataset. This dataset only contains the labels of each song with a series of features already extracted. The dataset itself contains neither the complete songs nor the genre labels of each song, but both can be downloaded through community-contributed works, such as "Last.fm dataset" found on MSD webpage [18], or "tagtraum genre annotations" [34, 35]. Interestingly, the genres used by this dataset are very similar to those of GTzan, merging Pop and Rock into one, replacing Disco music with Electronic music, and adding genres such as Latin music, Vocal music, and New Age. This dataset will be used in our work to validate the results of the CONEqNet network.

4.3 Training

For the training process, we have 1000 songs that are segmented in 30-second windows. The training process consists of 100,000 iterations. In each iteration, we use 800 random songs from the 1,000 that make up the dataset. At the end of the 100,000 iterations, the state of the network in which a better result of between 100,000 was achieved will not necessarily be the final state. We opted for such a high number of iterations since training the net is a very time-consuming process. In that case, we considered better training once for a high iteration count and then training multiple times for lower iterations.

Each iteration divides the training set (800 songs) into batches of 64 songs and executes the training of the model for each of the batches. Since we are using a high number of iterations, we can use lower learning rates for more precision, in this case 0.001.

For the validation of the neural network obtained in the learning process, songs from the MSD dataset were used, that is, the remaining 200 songs from the original 1000 that were not selected for the training. In Table 1, we can see the success rate of the model.

For some genres, such as classical music songs, the success rate reaches 93.2% success, but there are genres and genre groups that greatly reduce the average rate of up to 40.31% for Rock and Metal songs. As indicated above, the data used to verify the effectiveness of the network correspond to a subset of the MSD dataset. Some genres are shared by both GTzan and MSD, but for different genres, songs obtained by searching in the most popular songs section of Spotify have been used. In MSD, Pop/Rock and Jazz/Blues had been merged

Table 1 Success rate of the neural model designed

Musical genre	Success rate
Blues	54.46%
Classical	93.20%
Country	53.30%
Hiphop	50.16%
Jazz	87.34%
Metal	34.90%
Pop	85.426%
Reggae	73.88%
Rock	45.72%

into a single genre, so the remaining were obtained from Spotify by looking for "Rock" and "Jazz" playlists in the app. In total, 1,000 songs (100 for each genre) were used for the evaluation, both from MSD and Spotify. As can be seen later, these success rates will be sufficient to obtain the desired result, which is the automatic equalization of the songs. Most songs cannot be labeled as just a single genre, and most are a mixture of genres (or a mixture of characteristics common to different genres). In the datasets, songs are labeled as the most likely genre, so the classification may not be the same as the previous classification. In these cases, missing the label means that the song may not belong 100% to a single genre but to a mixture of them. In our strategy, we will use genre mixes, and we will consider equalization equally among those that the neural network determines to be the most likely.

4.4 Smart song equalization

As a proof of concept, the results of automatic equalization of a song will be displayed. In this test, the results are shown with the song "Gravity" of the musical group "Against the current". The song is divided into segments of 30 seconds, and for each segment, the output of the network that indicates the corresponding musical genre is obtained.

First, we will compare the classification outputs for the Softmax case against the Probabilistic case.

In Fig. 3, the results for the Softmax case can be observed. Both graphs represent the output of the neural model (y axis) for each time segment (x axis) corresponding to one genre or another. Each color corresponds to a genre according to its color. The first graph includes a representation of the individual output of each genre at each instant, while the second represents the outputs in a cumulative bar chart, where each genre appears alongside the others in each bar. The duration of the song used was 3 minutes and 42 seconds, so 7 fragments of 30 seconds each were generated, leaving the last 12 seconds unprocessed. In this proof of concept, the Softmax function is used to obtain the genres; the genres obtained for each segment are at a near 100% probability. For this song, there is clearly a structure in three parts, the first detected as Hip-hop, the second as Blues, and the end of the song between Classical and Blues. In the fifth segment (number 4), there is a small segment that corresponds to Country. This indicates that if we did not use the Softmax function, in the probabilistic method, the results for Classic and Country would be very similar.

If we look at the equalization profiles for Classical and Blues music (Figs. 4 and 5), it can be seen that, even if they are confused, they really respond to very similar equalization

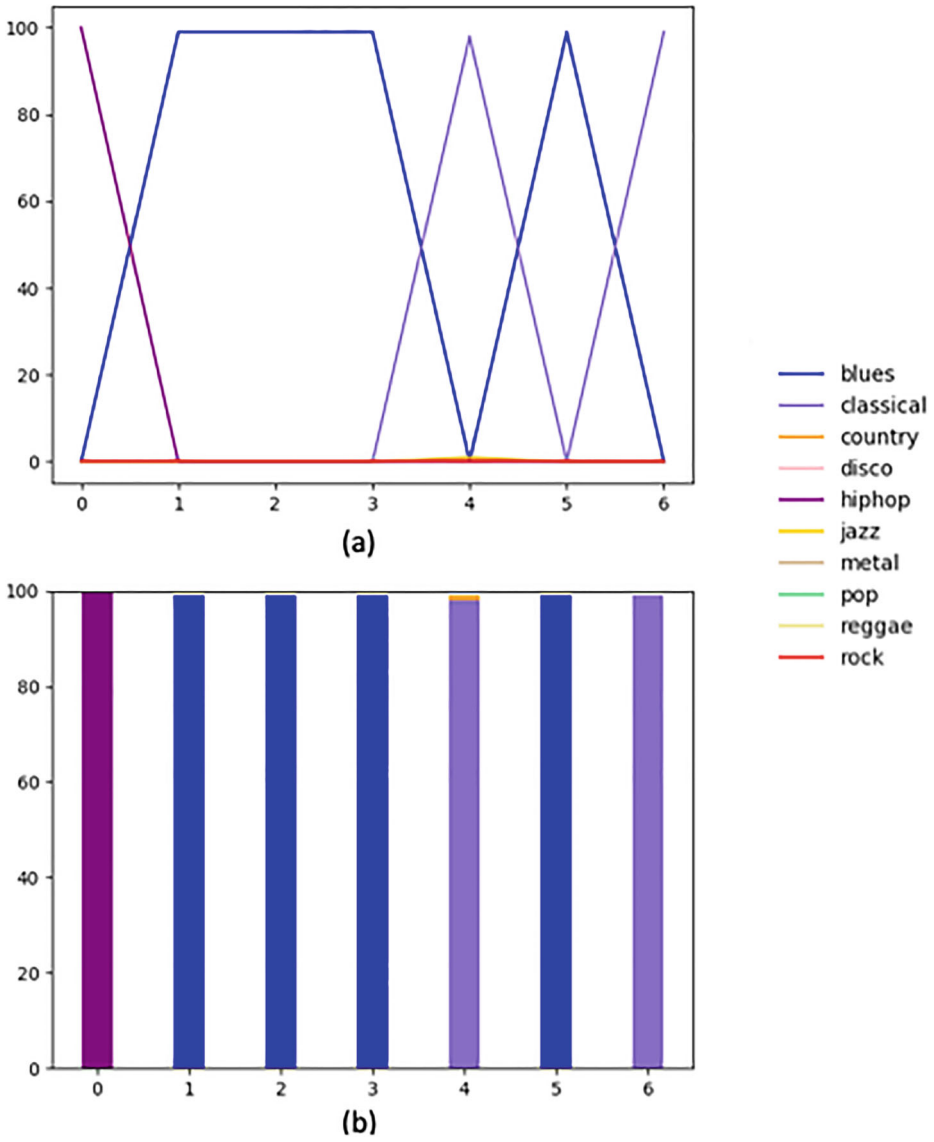

Fig. 3 Results of the song segment classification with the Softmax output: (**a**) in linear format, and (**b**) in bar format, as matching percentage (y-axis) over segment number (x-axis)

needs. If the objective is not to classify, but to equalize based on the classification, we can conclude that this is a good result.

A second proof of concept is performed with the same song but taking the probabilistic output. In this case, the three most probable genera for each segment were taken. It seems to be confirmed that the song follows the same structure described above, of three parts (Fig. 6) For the first segment, great confusion appears between Hip-Hop and Disco, accumulating

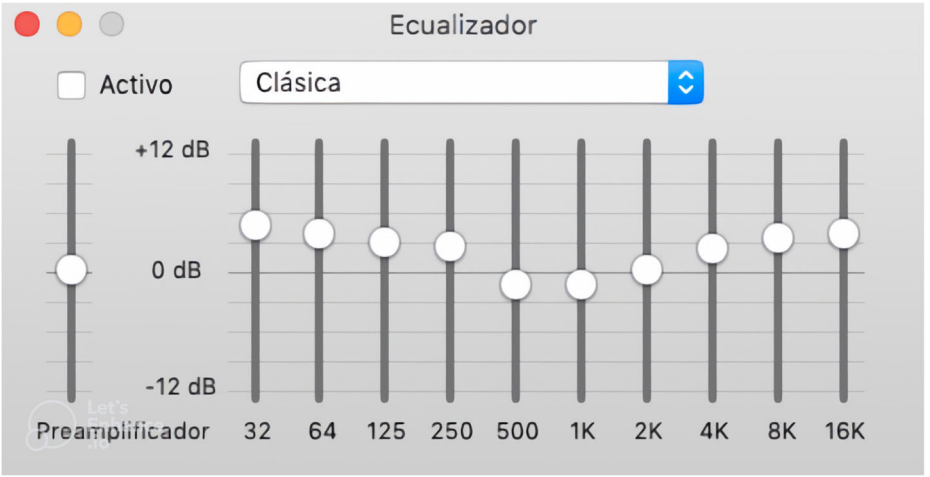

Fig. 4 Equalization profile for Classical music

between 75% of the probability. As in the case of Classical with Blues, both genres use relatively similar equalization profiles, especially in the intermediate and acute bands.

For the intermediate and final segments, important confusion occurs between Blues and Classical, as in the case of Softmax. Even if we compare the profiles of Blues, Classical, Hip-hop and Disco, the four genres follow a similar structure in the form of a 'V' that accentuates bass and treble but maintains or attenuates intermediate frequencies.

Finally, we want to know the results of automatic equalization if what we do is a weighted interpolation of the genres detected in each segment, which is the work intended in this paper. In Fig. 7, graphs represent the gain of each frequency band (y axis) with respect to each time segment (x axis). The bands are represented with colored lines, garnets and red in regard to high frequencies and blue in regard to low frequencies.

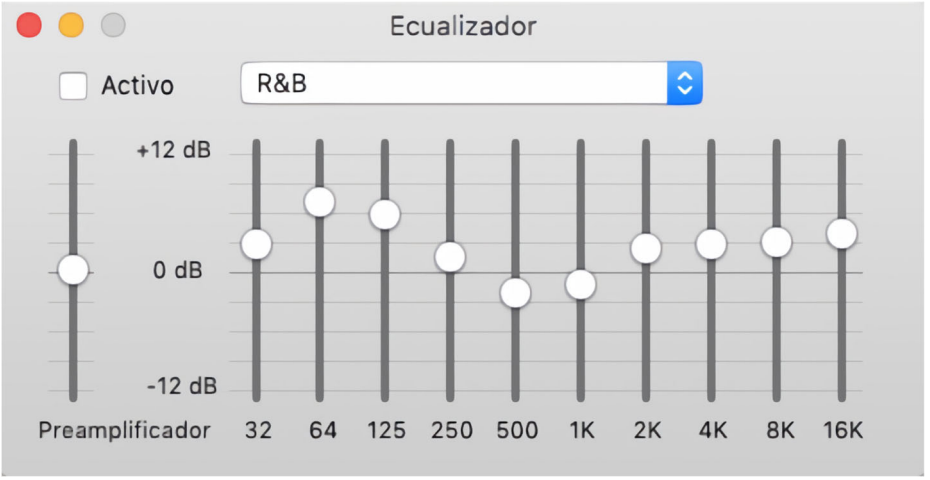

Fig. 5 Equalization profile for Blues music

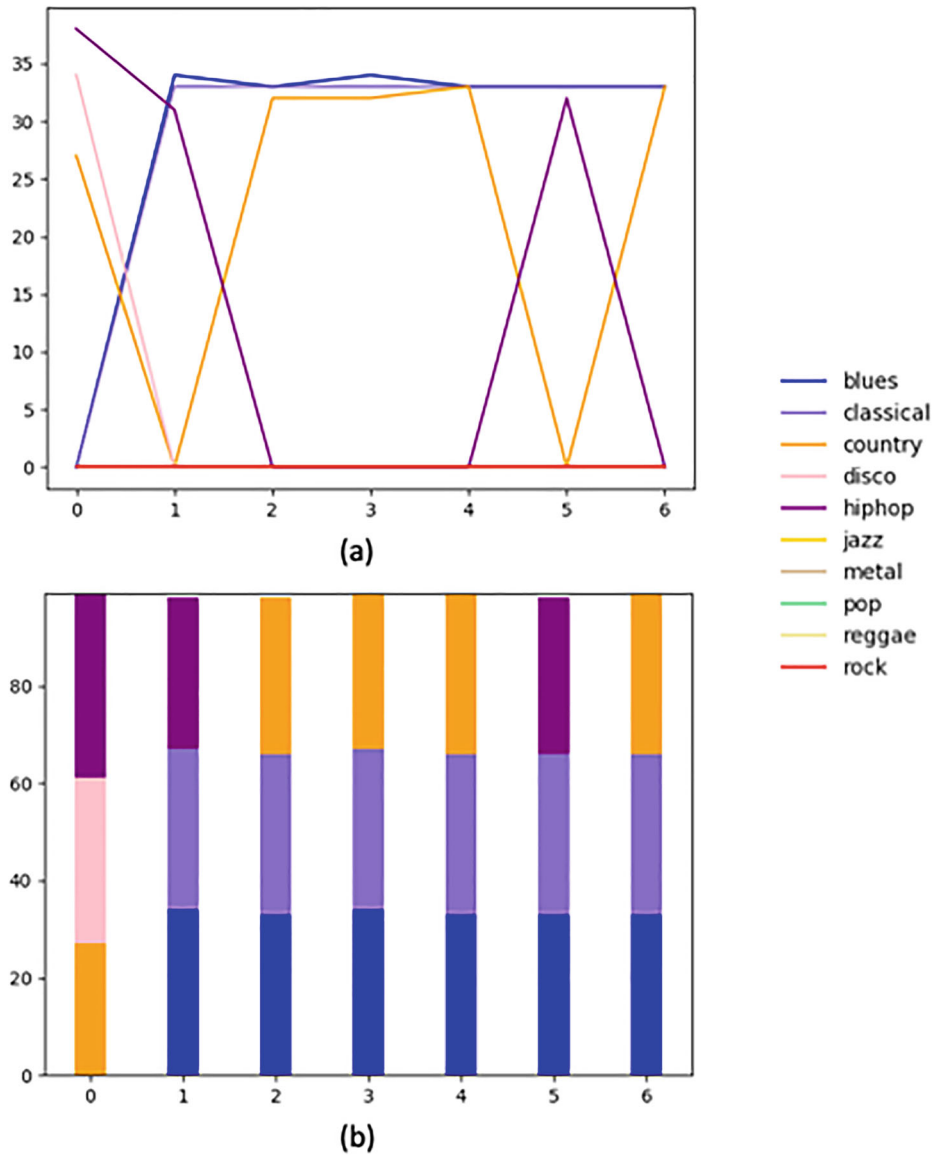

Fig. 6 Results of the song segment classification with the Softmax output: (**a**) in linear format, and (**b**) in bar format, as matching percentage (y-axis) over segment number (x-axis)

With this proof of concept, the suspicion of the structure of the song in three parts is confirmed. Comparing the interpolation system, in some moments, it seems that precision is lost; for example, between segments 0 and 1, the frequency of 2k appears above that of 8k (higher gain) in the second graph but appears below that in the first graph. Even so, applying interpolation is considered beneficial because it provides smoothness to transitions between segments.

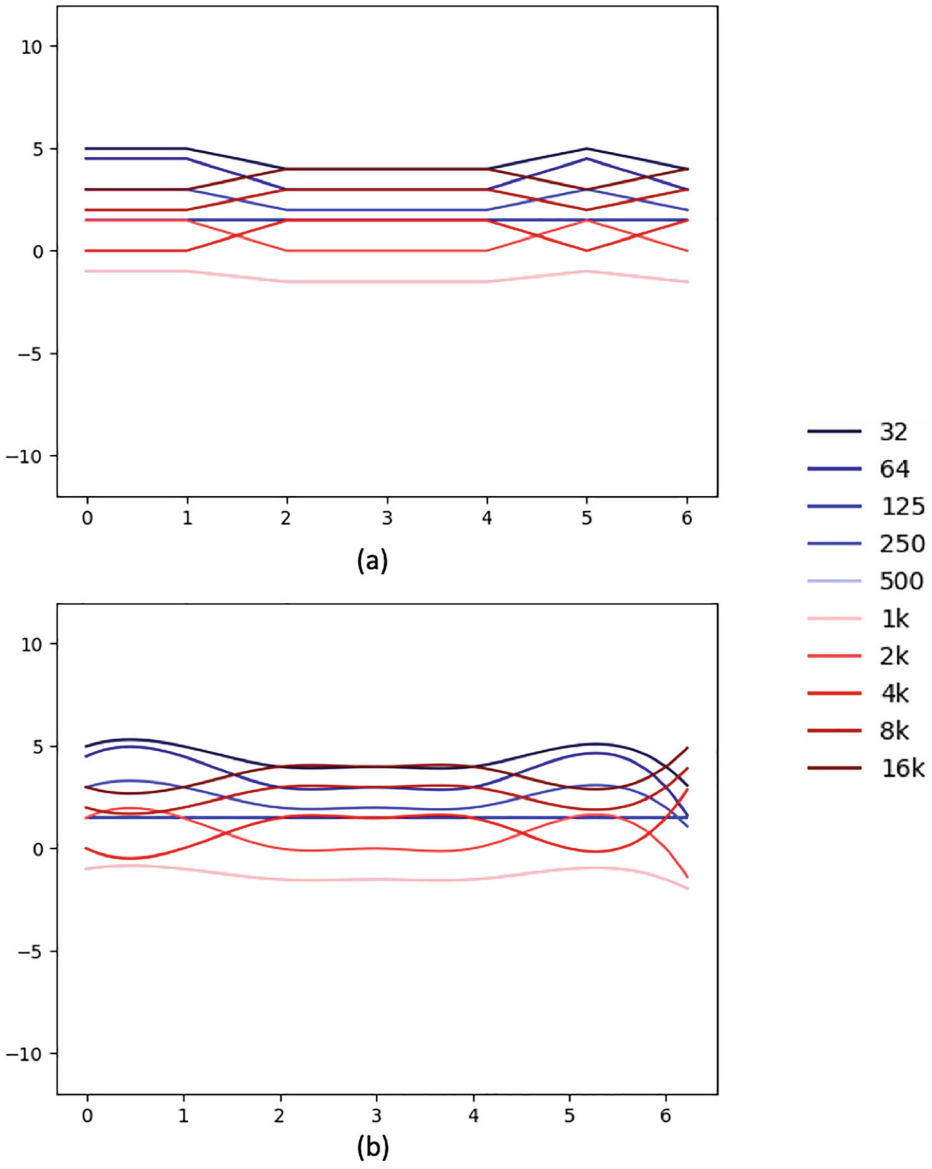

Fig. 7 Results of automatic equalization using a weighted mixture of the Softmax output: (**a**) without interpolation of genres; and (**b**) with interpolation of genres, as loudness differences in decibels (y-axis) over segment number (x-axis)

If we use the probabilistic output of the network, we can observe (Fig. 8) how the results obtained are very similar to those obtained by Softmax, although somewhat softer and less abrupt. This is the definitive confirmation that genre weighting not only works but is also beneficial, producing much cleaner and more effective results.

Table 2 A breakdown of the characteristics that have changed in each of the equalized fragments in *demo file*, separated in positive and negative changes

Song name	Demo time	Positive	Negative
Gravity	0:16-0:22	– Boosted bass, as It represents (with the voice) one of the main elements of this section	– Guitar sound obscured by the increase in volume in bass. Guitar is also one of the main elements as the song genre is Pop-Rock – No appreciable changes in the voice
– Gravity	0:28-0:34	– Clearer voice and voice harmonics – Louder bass	
Gravity	0:39-0:41	– More isolable and easier to differentiate instruments	– The song is too loud and overcrowded in this fragment
Dance with the devil	0:46-0:50	– Sharper guitars which stand out more than in the original song	
Dance with the devil	0:55-0:59	– More aggressive and fuller guitar sound, the main element in this fragment, with close to no loss in other instrument's presence	– The cymbals from the battery are too loud, instead of the accompaniment to the music they are meant to be
High Hopes	1:08-1:11	– The lower pitched brasswinds stand out way more, which results in a richer and more diverse sound	
High Hopes	1:15-1.19	– The brasswinds sound is increased to the levels of the voice. Both of them are the highlights of the song in this fragment	– The voice loses some protagonism, contrary to the intention the original song may have had
High Hopes	1:23-1:27	– In this segment in the original song, the higher pitched brasswinds stick out a little bit more, so they are further increased in volume in the equalization over the lower pitched ones	

For the reader to appreciate the improvements obtained with CONEqNet, the following link is provided to listen to different proofs of concept (use headphones for a better experience): Demo.

In this *demo file*, examples for three songs are shown: The previously analyzed "Gravity" from the musical group "Against the Current", "Dance with the Devil" from "Breaking Benjamin", and "High Hopes" from "Panic! at the Disco". These songs were chosen to represent a relatively wide spectrum of genres or musical qualities, using a song with clear sounds in the first case with "Gravity", typical in Pop/Pop-Rock music, a song with more distortion and aggressive sounds with "Dance with the Devil", which is common in metal music, and a more mainstream and processed sound in "High Hopes", which is slightly

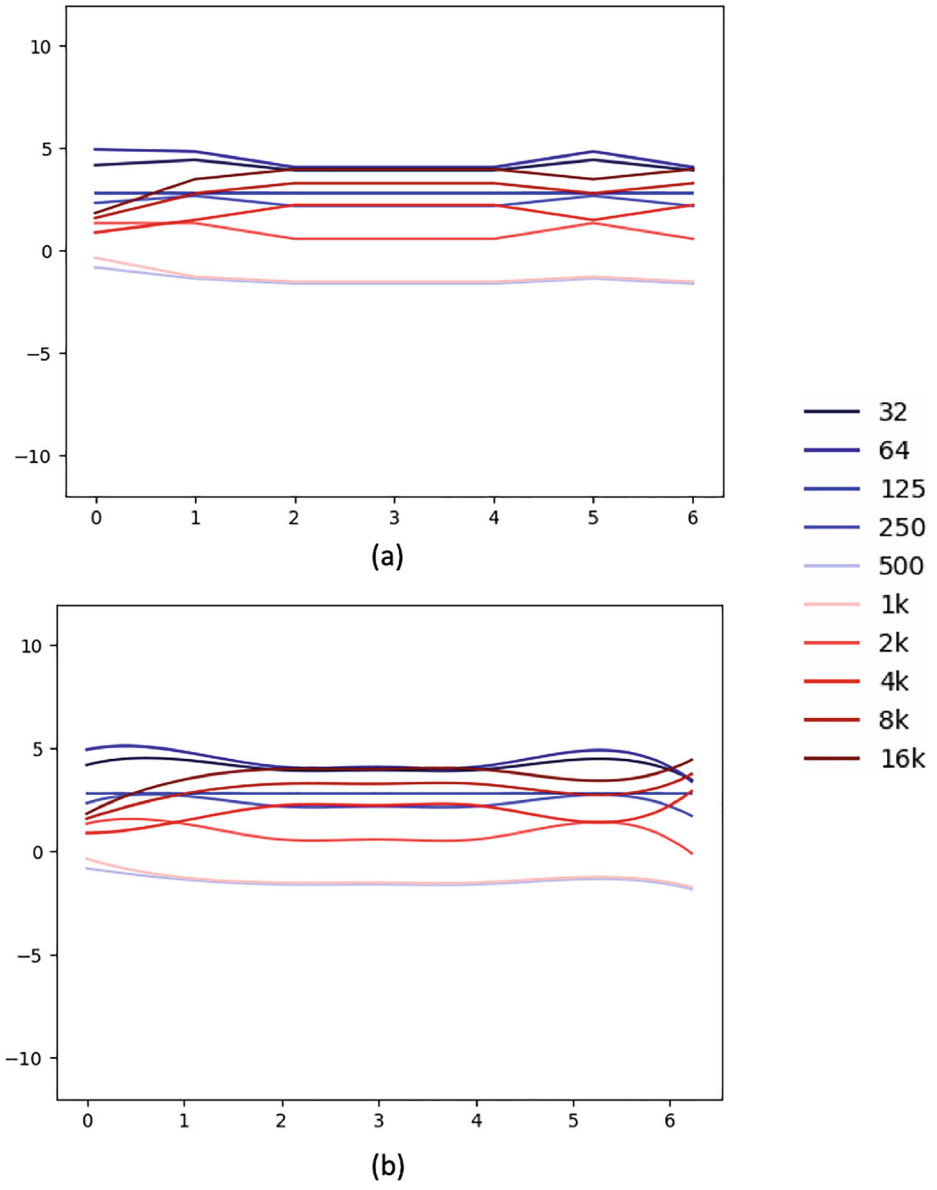

Fig. 8 Results of automatic equalization using a weighted mixture of the Softmax output: (**a**) without interpolation of genres; and (**b**) with interpolation of genres, as loudness difference in decibels (y-axis) over segment number (x-axis)

"poppier". Table 2 shows a breakdown of the characteristics that have changed in each of the equalized fragments, separated into positive and negative changes.

In the first case study, "Gravity", we found both positive and negative differences. The positive differences focus mainly on making some frequencies or instruments stand out

more, while the negative differences are based on obscuring some features by increasing the volume in others.

In the second case, "Dance with the Devil", we find that the differences are mostly positive, as they make the sound clearer, sharper and fuller.

Last, in the case of "High Hopes", the differences are minor. This can be due to the song being "poppier" and by that, more processed and with no need of additional processing.

It is important to note that in typical cases, no changes can be classified as either bad or good, as it is very hard to classify any musical element as bad or good. Some changes depend heavily on the subjectivity of the listener and cannot be evaluated in an objective way. Other changes answer a general necessity, either genre- or song-driven, to enhance certain features or instruments of a song, such as increasing the vocals volume on a pop song.

5 Conclusions

This paper presents a new intelligent parametric equalization model based on convolutional neural networks called CONEqNet. This model allows the equalization parameters to be adapted throughout the song. Moreover, taking into account that songs do not always belong to a specific musical genre, it performs an integration of the parameters of the musical genres that are being played at a given moment. The design of the model is based on the hierarchical nature of a piece of music, decomposing the process into three levels: low-level, mid-level, and upper-level features. The proof of concept tests verified the good results obtained with this new parametric equalization model.

The proposed architecture is a breakthrough in terms of obtaining an EQ profile by merging musical genres. This property will allow, for instance, to be able to equalize subgenres of music. For example, within classical music, there are subgenres such as a quartet or an opera. We can interpret each genre as a superposition of features. Classical music has its own set of common features that are present in all subgenres, such as a quartet or an opera. The way to differentiate the two subgenres is to label them with more than one genre; the system can detect a small presence of pop influence in an opera song, since both are very focused on vocal frequencies, and maybe a little bit of rock influence in a quartet, since both are based on string instruments.

This equalization model has been designed to evolve in future works by adding other sources of information to determine the equalization profile to be used at each moment of a song. One of the sources of information would be the preferred tastes of the users. By including the possibility of user feedback, the EQ profile would vary according to the user's preferences. The model can also be extended in the future to detect significant elements. The detection of these meaningful elements can be used to highlight a specific instrument or even an individual note from the rest of the sounds, regardless of genre. It would be possible to train a neural model capable of detecting these significant elements to be able to introduce this information into the integration of equalization profiles.

Another issue of interest is overfitting. This is often a problem intrinsic to any software optimization solution. If you use image enhancement software to improve the quality of an image over and over again, you will end up with a badly distorted image. The reason this

happens is because the system is trained to try to improve the quality of the image/song, not to distinguish whether it has already been improved or not. Future work will look at including a way to determine whether a song is already sufficiently equalized or not, and, accordingly, apply weaker or stronger modifications.

Acknowledgments This work was funded by the private research project of Company BQ, the public research projects of the Spanish Ministry of Science and Innovation PID2020-118249RB-C22 and PDC2021-121567-C22 - AEI/10.13039/501100011033, and the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17) and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Data Sharing Data sharing is not applicable to this article, as no new data were created or analyzed in this study. This work uses two public datasets (MSD-Million Song Dataset and GTzan) available at the http://millionsongdataset.com/ and http://marsyas.info/downloads/datasets.html, respectively.

Conflict of Interest Jesús Iriz, Miguel A. Patricio, Antonio Berlanga and José M. Molina declare that they have no conflict of interest.

References

1. Apple Inc. itunes
2. Bazgir O, Ghosh S, Pal R (2021) Investigation of REFINED CNN ensemble learning for anti-cancer drug sensitivity prediction. Bioinformatics:37
3. Bertin-Mahieux T, Ellis DPW, Whitman B, Lamere P (2011) The million song dataset. In: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011
4. Bohn DA (1988) Operator adjustable equalizers: An overview. In: Audio engineering society conference: 6th international conference: Sound reinforcement
5. Casagrande N, Eck D, Kégl B (2005) Geometry in sound: a speech/music audio classifier inspired by an image classifier. In: International Computer Music Conference (ICMC)
6. Chen J-H, Su M-C, Azzizi VT, Wang T-K, Lin W-J (2021) Smart Project Management: Interactive Platform Using Natural Language Processing Technology. Appl Sci 11(4)
7. Cheng YH, Chang PC, Nguyen DM, Kuo CN (2021) Automatic music genre classification based on crnn. Eng Lett 29(1)
8. Choi YJ, Rahim T, Nyoman Apraz Ramatryana I. (2021) Improved CNN-based path planning for stairs climbing in autonomous UAV with liDAR sensor. In: 2021 international conference on electronics, Information, and Communication (ICEIC)
9. Dieleman S Recommending music on spotify with deep learning. https://benanne.github.io/2014/08/05/spotify-cnns.html
10. Elbir A., İlhan HO, Serbes G, Aydın N (2018) Short Time Fourier Transform based music genre classification. In: 2018 Electric electronics, computer science, biomedical engineerings' meeting, Istanbul, pp 1–4

11. George J, Shamir L (2015) Unsupervised analysis of similarities between musicians and musical genres using spectrograms. Artificial Intelligence Research
12. Goel A, Sheezan M, Masood S, Saleem A (2015) Genre classification of songs using neural network. In: Proceedings - 5th IEEE International Conference on Computer and Communication Technology, ICCCT 2014
13. Hargreaves DJ, North AC (1999) The functions of music in everyday life: Redefining the social in music psychology. Psychol Music 27(1)
14. Hossan MA, Memon S, Gregory MA (2010) A novel approach for MFCC feature extraction. In: 2010 4Th international conference on signal processing and communication systems, pp 1–5
15. Kaur C, Kumar R (2017) Study and analysis of feature based automatic music genre classification using Gaussian mixture model. In: 2017 International conference on inventive computing and informatics (ICICI), pp 465–468
16. Khonglah BK, Mahadeva Prasanna S. R. (2016) Speech / music classification using speech-specific features. Digital Signal Process Rev J 48:71–83, 1
17. Kundu P, Kundu P, Mallik S, Bhowmick S, Mandal P, Banerjee H, Pal SB (2022) Facial expression recognition using convoluted neural network (CNN). In: Lecture notes in networks and systems, vol 291
18. last.fm. The official song tags and song similarity collection for the million song dataset. http://millionsongdataset.com/lastfm/
19. Li T, Choi M, Fu K, Lin L (2019) Music sequence prediction with mixture hidden markov models. In: IEEE International conference on big data (big data), Los Angeles, pp 6128–6132
20. Liu C, Feng L, Liu G, Wang H, Liu S (2021) Bottom-up broadcast neural network for music genre classification. Multimed Tools Appl 80(5)
21. Liu C, Wei Z, Ng DWK, Yuan J, Liang Y-C (2020) Deep transfer learning for signal detection in ambient backscatter communications. IEEE Trans Wirel Commun 20(3):1624–1638
22. Mandal P, Nath I, Gupta N, Madhav KJ, Dev GG, Pal S (2020) Automatic music genre detection using artificial neural networks In: Intelligent Computing in Engineering. Springer, Singapore, pp 17–24
23. Narkhede N, Mathur S, Bhaskar A (2022) Automatic classification of music genre using SVM. In: Computer networks and inventive communication technologies. Springer, pp 439–449
24. Narkhede N, Mathur S, Bhaskar A (2022) Machine learning techniques for music genre classification. In: Information and communication technology for competitive strategies (ICTCS 2020). Springer, pp 155–161
25. Patel R, Patel S (2021) Deep learning for natural language processing. In: Lecture notes in networks and systems, vol 190
26. Prabhu NR, Andro-Vasko J, Bein D, Bein W (2018) Music genre classification using data mining and machine learning. In: Latifi S (ed) Information technology - new generations. Springer International Publishing, Cham, pp 397–403
27. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1988) Numerical recipes in C: the art of scientific computing. Cambridge University Press, USA
28. Qiu L, Li S, Sung Y (2021) Dbtmpe: Deep bidirectional transformers-based masked predictive encoder approach for music genre classification. Mathematics 9(5)
29. Rentfrow PJ (2012) The role of music in everyday life: Current directions in the social psychology of music. Soc Personal Psychol Compass 6(5):402–416
30. Santika IKG, Sa'adah S, Yunanto PE (2021) Gold price prediction using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
31. Savchenko AV, Demochkin KV, Grechikhin IS (2022) Preference prediction based on a photo gallery analysis with scene recognition and object detection. Pattern Recogn:121
32. Srinivasa Murthy YV, Koolagudi SG (2018) Content-based music information retrieval (CB-MIR) and its applications toward the music industry. A Rev ACM Comput Surv 51(3):6
33. Sturm BL (2012) An analysis of the GTZAN music genre dataset. In: MIRUM 2012 - Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, Co-located with ACM Multimedia 2012
34. Schreiber H (2015) Improving Genre Annotations for the Million Song Dataset. In: Proceedings of the 16th International Society for Music Information Retrieval conference, Málaga, pp 241–247. ISMIR
35. Tagtraum Industries. Tagtraum genre annotations for the million song dataset. https://www.tagtraum.com/msd_genre_datasets.html
36. Tjoa S Notes on Music Information Retrieval. Mel Frequency Cepstral Coefficients (MFCCs). https://musicinformationretrieval.com

37. Tzanetakis G, Cook P (2002) Musical genre classification of audio signals. IEEE Trans Speech Audio Process 10(5):293–302
38. Välimäki V, Reiss JD (2016) All About Audio Equalization: Solutions and Frontiers. Appl Sci 6(5)
39. Yue Z, Gao F, Xiong Q, Wang J, Huang T, Yang E, Zhou H (2021) A Novel Semi-Supervised Convolutional Neural Network Method for Synthetic Aperture Radar Image Recognition. Cogn Comput 13(4)