UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Master Universitario en Deep Learning for Audio and Video Signal Processing

MASTER THESIS

Deep Learning based singer identification

Laura Bustos Manzanet
Advisor: David Martín Gutiérrez
Lecturer: Daniel Ramos Castro

September 2021

Deep Learning based singer identification

AUTOR: Laura Bustos Manzanet
TUTOR: David Martín Gutiérrez

AUDIAS
Depto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre de 2021

Resumen

Es sabido que la identificación de locutor es un campo con mucha investigación relacionada llevada a cabo, pero cuando se trata de buscar investigación desarrollada a partir de voz cantada solo existen unos pocos estudios. Esta diferencia en la cantidad de trabajo relacionado con ambos campos se debe mayoritariamente a que la voz hablada es más simple y contiene un espectro de frecuencia mucho más estrecho que la voz cantada. De esta manera, el presente Trabajo Fin de Máster contiene un estudio para identificar cantantes a partir de sus canciones grabadas. Para ello, se ha desarrollado un sistema más sofisticado para paliar el aumento de complejidad en los datos, que sea capaz de discriminar entre cantantes.

Como paso previo a la identificación de cantante, y debido a la evidente escasez de bases de datos de voz cantada en el estado del arte, el presente trabajo también desarrolla un método automático de generación de una novedosa base de datos mediante el uso de la API de Spotify. La base de datos presenta información de género musical, artista y diferentes características acústicas de los extractos de audio de 30 segundos que utiliza Spotify como pre visualización. Los archivos de las canciones se han separado con la red de la aplicación de Spleeter para llevar a cabo una separación de fuente y así poder trabajar con el archivo obtenido que solo contiene la voz cantada de las canciones originales.

El sistema desarrollado se ha basado en diferentes extractores de características del estado del arte utilizando tanto técnicas de análisis de voz hablada como de identificación de instrumentos musicales. Con las características obtenidas, se han alimentado varios clasificadores del estado del arte basados tanto en redes neuronales poco profundas como redes de identificación de locutor o voz hablada.

Palabras clave (castellano)

Identificación de cantante, Aprendizaje Profundo, wav2vec, transformada scatter, canciones, separación de fuente, APIs, Spotify, x-vectors, Spleeter, base de datos, MIR, aprendizaje por refuerzo

Abstract

It is known that speaker identification is a field with a lot of related research carried out but, when it comes to looking for research developed from singing voice instead of speech, only a few studies can be found. This difference in the amount of work related to both fields is mainly due to the fact that the spoken voice is simpler and contains a much narrower frequency spectrum than the singing voice. In this way, this Master's Final Project contains a study to identify singers from their recorded songs. For this purpose, a more sophisticated system has been developed to face the increased complexity in the data, being able to discriminate among singers.

As a previous step to identify the singer, and due to the scarcity of databases of singing voice in the state of the art, the present work also includes the development of an automatic way for creating a novel database using Spotify's API. The database contains information related to the musical genre, the artist and different musical characteristics of the 30 seconds excerpt pre-view song provided by Spotify. The files of the songs have been source separated with the network of the Spleeter application to carry out a source separation and thus be able to work with the processed file that only contains the singing voice of the original songs.

The developed system has used different feature extractors from the current state of the art using both speech analysis techniques and techniques that are used when musical instruments are wanted to be identified in recordings. With these obtained features, some current state of the art classifiers have been fed based on shallow neural networks and speaker identification networks.

Keywords

SingerID, Deep Learning, wav2vec, scatter transform, songs, source separation, APIs, Spotify, x-vectors, Spleeter, database, MIR, transfer learning

Acknowledgements

En primer lugar, me gustaría dar las gracias a David, mi tutor, por su ayuda y sus consejos hasta el último momento, los cuales han sido claves en el transcurso de este proyecto. Además, la ayuda de Joaquín y Alicia ha dado otro punto de vista a los problemas y las soluciones gracias a su amplia experiencia y su paciencia y dedicación a la hora de enseñarme.

Gracias al grupo de investigación AUDIAS de la EPS por acogerme en él y con sus charlas de investigación semanales, introducirme en el amplio y variado mundo de la investigación científica.

Gracias a Alejandro, mi luz y mi apoyo incondicional en cualquier momento. La sorpresa más bonita que me podría haber dado Madrid.

Gracias a Luis y a Héctor por motivarme a realizar el trabajo en el mundo de la música, que es su vida.

Por último, pero no menos importante, agradecerles a mis padres, hermana y amigos todo lo que me han dado y su constante apoyo.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1 Introduction

1.1 Motivation

The task of tagging songs in mostly all of the streaming platforms like Spotify or YouTube is done semi-automatically introducing one by one all the information related to a song and its artist in a recursive way since now. With the latest technologies, these platforms are starting to use AI for automatic tagging. In this project we will approach a solution to the problem by building a system able to identify singers in songs and thus convert the task from a people and time-consuming task to an easy and fast one.

This is a new challenging problem as there are many algorithms from the state-of-the-art literature tested to identify speakers with deep learning techniques in clean audio but not in more realistic scenarios like audio files with noise or with low quality. The fact of having clear audios of speech is really strange and in this project the identification of the speaker in songs has been developed. As the speech is a part of a song, the audio file that will be analyzed does not only have the issue of having background noise but also the change of the acoustic features related to the voice itself. This last point is expected to be tricky as the current solutions to speaker identification problem using MFCC (Mel Frequency Cepstral Coefficients) or i-vectors and x-vectors as ensemble of the features representing the speaker depend on the acoustic characteristics of the voice. In some cases, this would be an incorporated difficulty as they have not been tested with the acoustic characteristics of the singing voice of a person. We will implement and analyze the performances of the most spread and famous algorithms from the state-of-the-art of the speaker identification problem applied to our data.

A specific database will be built containing the information of the singers of the songs from the Spotify API into a text file and a 30-seconds preview of its top 5songs on Spotify in two audio files, one containing the whole song and the other containing only the vocals. This second file will be obtained using a high-quality state-of-the-art system used to source separate the parts of a song into the vocals part and the background music part. Experiments are conducted with both sources of audio files, the original and the vocals one, and compared between them.

1.2 Objectives

The main objective of this project is the development of a singer identification system with deep learning techniques. For that purpose, a state-of-the-art research has been proposed to be done in the field of the speaker identification problem using deep learning techniques and their adaptation to the analysis of the identity of the speaker but within a song. As it is a novel approach to the problem, it has been investigated in the field of techniques that have been demonstrated to be useful in the speaker identification problem. Also, the familiarization with the API of Spotify required to build the database is also required. Once

these two points are covered, the systems chosen will be implemented and their performances will be evaluated adapted to our specific data.

The main objectives proposed in this thesis can be summarized as follows:

1. Generation of the database containing both previews of songs available in Spotify and their associated vocals stems.
2. Analysis of the State-of-the-Art Speaker Identification techniques using deep learning that could be applied to our problem.
3. Development of the systems of each of the procedures chosen from the previous point.
4. Running of the experiments and summarize the results.
5. Analysis of the results; we expect to be able to draw conclusions on:
    a. The performance of each system proposed.
    b. The fit of the systems to our data.
    c. Apply Transfer learning using the proposed dataset.

1.3 Structure of the report

This report has the following chapters:

- **Related work**
- **Design**
- **Development**
- **Integration and experimental results**
- **Conclusions and future work**

2 Related work

2.1 Database

Public Datasets for Singer Identification are not available nowadays. This problem is mostly due to the copyright of the songs. So, the researches done in this field work with specific databases created by the authors of the experiments. In [1] 20 male singers are recorded singing and reading 30 passages of 17-26 seconds of Mandarin pop songs. In [2] one album from each of the selected 10 singers (5 males and 5 females) from China are used. In [3] a database of 13 male and female performers with varying levels of singing skills is built. Each of these singers has 4-6 melodies of 20-30 seconds of duration. In [4] 10 male and 10 female Chinese singers were chosen with 30 songs from each of them. In [5] 50 songs of each of the 6 male and female Indian singers chosen constitute the database. In [6] 17 solo singers and 200 songs from the NECI Minnowmatch testbed were chosen. Finally, in [7] five popular Hindi songs of seven different singers with a duration of 50 seconds have been recorded. As a conclusion, it can be seen that the datasets of the state-of-the-art have a lot of drawbacks such as: small number of singers, not available universal audio files, musical genre bias and song style bias. For that reason, a novel dataset has been created.

2.2 Singer Identification without deep learning

The singer identification framework does not contain a great amount of related experiments having been developed, but it is clear that most of this work is out of the field of deep learning.

If we want to analyze the work, we clearly see that there are two parts in all of the systems developed: the feature extraction and the classification techniques used to differentiate among the different singers of the databases.

2.2.1 Feature extraction

For the extraction of the features of the singing voice, we can see that two different families of features are usually studied: the ones related to the Mel-scale and the ones which use another frequency scale.

The MFCC [1][2][3][5], Mel-Frequency Cepstral coefficients, are the most successful acoustic features in speech, speaker recognition, artist identification and instrument identification [3]. They carry less information on pitch than vocal tract configuration so they should be able to absorb discrepancy between singing and speech in the pitch variations [1]. The other Mel-scale coefficients are the LPMCC [2], Linear Prediction Mel-frequency Cepstral Coefficients, which are superior to LPC and MFCC in the human auditory sense to improve the efficiency of the human auditory characteristics.

GTCC [2], GammaTone-filterbank Cepstrum Coefficients, are log magnitude DCT of the power spectrum, also called delta and delta-delta coefficients, which model the cochlea by a bank of overlapping bandpass filters. They are a widely used model of auditory filters in the auditory system to simulate the motion of the basilar membrane within the cochlea as a function of time, in which the output of each filter models the frequency response of the basilar membrane at a single place. The MDCT [4], Modified Discrete Cosine Transform, is used as its lower coefficients of the DCT represent a rough shape of the spectrum [3]. Moreover, other coefficients widely used are the LPC [5], Linear Prediction Coefficients. These coefficients model the vocal tract response using a time-varying all-pole filter function which corresponds to the formants, resonances, of the vocal tract. A variation of these coefficients is called WLPC [6], Warped Linear Prediction Coefficients, which are a kind of LPC, but with pre-warping the power spectrum of each frame, which emphasizes on lower frequencies to pick out individual low harmonics. Finally, the audio features like the timbre, the pitch class and the loudness [5] are also used in combination with the previous explained coefficients in order to enrich the model of the features extracted from the singing voice audio files.

2.2.2 Classification

In order to classify the audio files, there is a list of techniques used to cluster the feature vectors of each singer together. All of the studies in this field share the point that they use a GMM [1][2][3][4][5][6], Gaussian Mixture Model. This model is based on multiple weighted Gaussians which are flexible to be adapted to the distributions not well modelled by a single cluster being able to encompass almost any distribution of data. The parameters of the gaussians are optimized via EM, Expectation Maximization, which is an iterative algorithm. In addition to the GMM, other methods are built in order to compose more complex classification systems. In [3] the KL, Kullback-Leibler, divergence is used in addition to the GMM to build a more robust classifier. Also, a K-NN, K-Nearest Neighbors, algorithm is used in [4] and [5] to optimize the clustering by computing the distances between each sample that is wanted to be classified and the cluster to which the k nearest samples are assigned. In [6] a SVM, Support Vector Machine, classifier is designed as the computation of an optimal hyperplane that can separate two classes of data based on statistical error minimization techniques. To lighten the data that these classifiers handle, in [5] and [6] a PCA, Principal Component Analysis, is done as a pre-processing step in order to normalize the data variances and reduce the size of the data used to build the classifiers.

2.3 Singer identification with deep learning

As has been stated previously, there are only two projects in the state-of-the-art of singer identification using deep learning techniques. Both systems have the same parts as in the case of the non-deep learning studies. The architectures developed are based on a feature extraction step followed by an artificial neural network as a classifier of the features extracted from the data.

2.3.1 Feature extraction

The features extracted in both cases fusion the MFCCs, as in the case of non-deep learning approaches, with musical features which tend to enhance the characterization of the singing voices. In [4] a mix of the MFCC is computed in addition to the timbre (the tone color or tone quality), the pitches (auditory sensation in which a listener assigns musical tones to relative positions) and the loudness (quality of a sound which is a primary psychological correlate of physical strength) of each audio. However, in [6] more musical features are computed.

These features are:

- event density (average frequency of events)
- pulse clarity (amount of rhythmic clarity or strength of the musical beats)
- spectral irregularity (degree of variation between successive peaks)
- zero cross rate (number of times the signal crosses the X-axis and identifies percussive sounds)
- spectral centroid (weighted mean of the frequencies component of the signal)
- skewness (measure of the symmetry or asymmetry nature of the signal)
- kurtosis (determines the noisiness nature of the signal)
- Shannon entropy (shows if the signal contains predominant peaks or not)

The main purpose of computing more musical features is to model the complex singing voice signal better.

2.3.2 Classification

In the classification step, shallow neural networks are used to classify inputs into a defined group of targets. It is a method based on gradient descent and it is used to compute the gradient of the error function with respect to all of the weights in the neural network and they are modified to attempt to minimize this error. In [4] a simple neural network made of one input layer, one hidden layer with 50 neurons and one output layer is used. Moreover, in [6] a feed-forward network of two layers with hidden sigmoid and softmax output neurons is built. The size of the hidden layer is set to 30 and 40 neurons in order to analyze which one performs with this type of data better. In addition to these layers, the four multilayer training functions Levenberg - Marquard (LM), Bayesian Regularization (BR), Scaled Conjugate Gradient (SCG) and One-Step Secant back-propagation (OSS) are used to classify the musical features computed previously from the singing audio files.

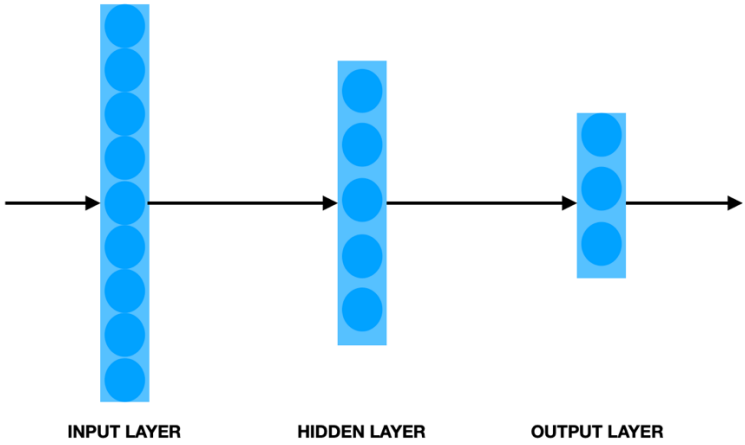

Figure 2-1: Neural Network

3 Design

Our system will follow the same structure as the proposed methods from the state-of-the-art with three main blocks as can be seen in the image 3-1: the database, the feature extractor and the classifier. The aim of this design is to build a generic and realistic system able to deal with any type of singing audio file without spending too much time and reducing the computing cost.

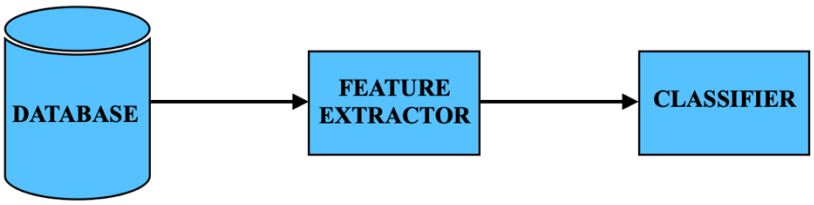

Figure 3-1: System architecture

This section is divided into two parts. The first one is the dataset that has been developed to be able to work in the identification of a singer from its recorded songs. Then, the evaluation process of the developed system is described. This process will be used to analyze the performance of the architecture.

3.1 Database

As we have seen, the databases used in the state-of-the-art projects so far have a lot of characteristics that make them poor databases. Some of the constraints they have are listed below:

- gender biased
- small number of speakers
- few amounts of data
- only one language/accent
- one specific song or musical genre.

In addition, they are not public datasets as they have problems with the copyright of the songs or are recorded by the researchers themselves.

So, in order to solve this big problem, a singer identification dataset has been created to be able to work in the singer identification/verification field. To build this dataset we have used the Spotify API to collect information about the songs. This is a very useful API as it contains mostly all of the songs in the world nowadays with a lot of information not only regarding the singers but also musical features of the songs. The database is composed by:

- 200 female singers
- 5 musical genres: Pop, Rock, Jazz, R&B and Indie
- 5 chunks of songs of 30 seconds per singer
- 9 musical features per song
- 2 stems of the songs: original and the vocals

As it is known, the speaker identification techniques nowadays have shown that the task of identifying speakers is much more accurate with male voices than female voices. This fact may be due to the wider frequency range of the female voice and its location in higher frequency spectrum than the male voices. So, our database contains the voices of 200 female singers to face the identification from the hardest point of view.

The dataset is practically balanced among the 5 most popular musical genres: Pop, Rock, Jazz, R&B and Indie. The 40 singers with the largest amount of available songs of each genre has been chosen as can be seen in figure 3-2. With this election we tend to have a robust database which will not be overfitted to a genre or a specific song as the state-of-the-art datasets are.

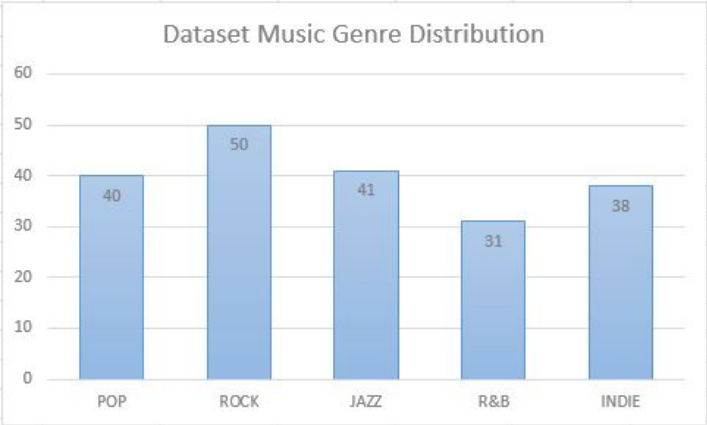

Figure 3-2: Dataset music genre distribution

With all of the information related to each song of the database, a csv file has been done to be able to access easy and fast to this information and, to evaluate the performance of the systems depending on not only the singer but also the musical genre and some musical features provided by Spotify's API. Each of the songs of the dataset contain the following information:

- Musical genre
- Name of the artist
- Name of the album
- Name of the song

- ID of the song in Spotify API
- URL of the song preview in Spotify API
- Musical features provided by Spotify API: acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence

As the main problem when you need to work with music is its copyright, the database with the song audio files that has been created is based on the preview files provided by Spotify's API. Each of these files are 30 seconds extracts of the songs around the chorus of the song stored in the database as original songs. Moreover, the Spleeter [22] code developed by Deezer has been used to source separate the songs. This code splits the song in two files: vocals and background music. The vocals audio file is the most useful file to work in the singer identification field as it only has the information of the voice and mutes the musical background. The Spleeter code has been the source separation chosen code since it does not provide a perfect source separation but a really accurate one that will be useful for our purpose.

3.2 Evaluation metrics

In order to evaluate the classifiers, it is necessary to quantify their performances. Thus, there is a big number of different criteria and values but in this project, we will be centered in the accuracy and the Cross-entropy loss of the developed neural networks.

3.2.1 Accuracy

The accuracy is the number of correct predictions with respect to the total number of samples. In order to compute it, we have followed two different approaches. This decision has been done as the last experiment developed is based on the utilization of a pre-trained network to verify if songs are related to the same singer and not in the common neural network train-test procedure.

For the first six experiments are based on training and testing networks. The accuracy of the performances of the systems has been computed as the total number of correct predictions, singer classes predicted that match the singer labels of the dataset, out of the total number of samples processed by the networks as can be seen in equation I.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \qquad (\text{I})$$

As the experiment 7 only performs inference with the pre-trained network, the confusion matrix has been used to compute the accuracy. This matrix provides information of the performance of the classifier through the different hypothesis that each prediction could match. These situations can be: TP (true positive), a real positive sample is detected as positive; TN (true negative), a real negative sample is detected as negative; FP (false

positive), a real negative sample is detected as positive; and FN (false negative), a real positive sample is detected as negative.

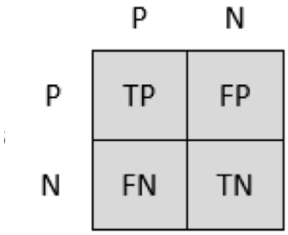

Figure 3-3: Confusion Matrix

If we use the values obtained from the confusion matrix that can be shown above, the accuracy can be computed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (\text{II})$$

3.2.2 Cross-entropy loss

Loss functions are used to measure the error between the prediction output of the network and the provided target value used as label of the sample. A loss function tells us how far the algorithm model is from realizing the expected outcome computing the penalty that the model gets for failing to yield the desired results.

In this case, the Cross-entropy loss has been chosen. This criterion combines LogSoftmax and NLLLoss loss functions in one single class. It is useful when training a classification problem with *C* classes as the one we are facing trying to classify audio files among 200 singers. It is used to work out a score that summarizes the average difference between the predicted values and the actual values. To enhance the accuracy of the model, you should try to minimize the score. The cross-entropy score is between 0 and 1, and a perfect value is 0. Moreover, this function penalizes greatly for being very confident and wrong, giving a great penalty to both incorrect but confident predictions and, to correct but less confident predictions.

The loss can be described as:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right) \qquad (\text{III})$$

In addition, if we have a dataset with unbalanced classes, we can balance the loss of the network by setting a weight to each class and compute the loss as:

$$\text{loss}(x, class) = weight[class] \left(-x[class] + \log \left(\sum_j \exp(x[j]) \right) \right) \qquad \text{(IV)}$$

The losses are averaged across observations for each minibatch in order to provide a loss value per batch of observations as follows:

$$\text{loss} = \frac{\sum_{i=1}^{N} loss(i, class[i])}{\sum_{i=1}^{N} weight[class[i]]} \qquad \text{(V)}$$

This loss function will help us providing an accurate prediction with a higher probability.

4 Development

In this section, the blocks of the developed system's architecture explained in the previous chapter are explained. The aim of each of them has been to adapt better the deep learning classification pipeline to our data while maintain the guidelines marked in the state-of-the-art systems.

4.1 Database

The first part, the database, has been explained in detail in the previous section.

4.2 Feature extraction

The second part of the project is the feature extractor. The audio files should be pre-processed before feeding the network. In this step, we will build a model of the audio files that will compress the information of the voice and its frequential characteristics. Our model will describe the singing voice as a mixture of speech and music. To this aim, we have used representation of the song containing: x-vector obtained from a state-of-the-art x-vectors extractor for the speech point of view and the coefficients of the Scatter transform in order to extract the frequential features of each singing voice.

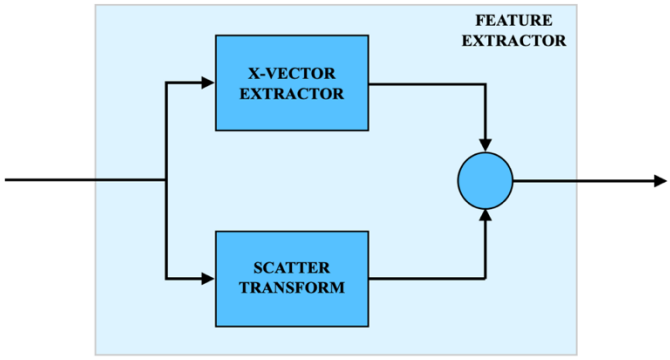

Figure 4-1: Feature extractor

4.2.1 X-vector extractor

In order to represent the signal as speech signals, we have chosen the x-vector. This type of representation aims to replace feature vectors for text-independent speaker verification with embeddings extracted from a feedforward deep neural network. Long-term speaker characteristics are captured in the network by a temporal pooling layer that aggregates over the input speech. This enables the network to be trained to discriminate between speakers from variable length speech segments. After training, utterances are mapped directly to fixed-dimensional speaker embeddings and pairs of embeddings are scored using a PLDA-based backend. It is demonstrated that the embeddings outperform i-vectors for short speech

segments and are competitive on long duration test conditions. Prior studies have found that embeddings leverage large-scale training datasets better than i-vectors.

We have chosen as x-vector extractor the wav2vec 2.0 [14]. This is a framework for self-supervised learning of speech representations that masks latent representations of the raw waveform and solves a contrastive task over quantized speech representations as can be seen in the figure below.

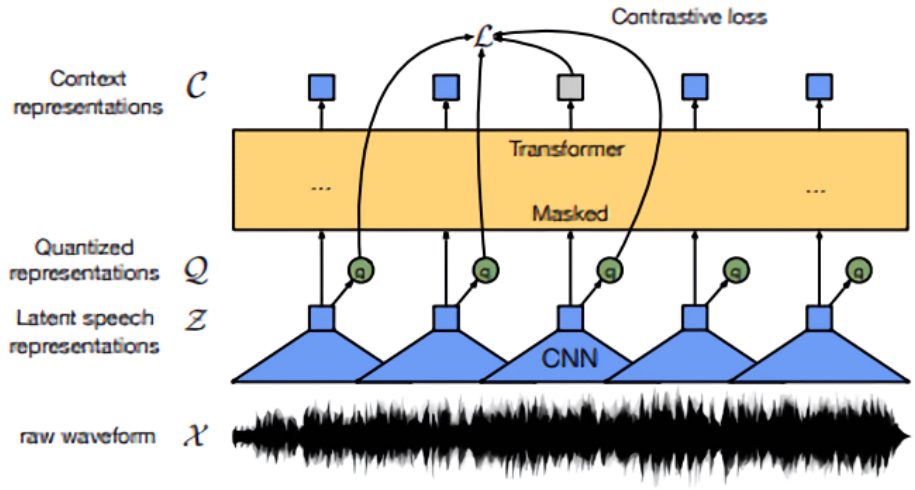

Figure 4-2: Wav2vec 2.0 system [14]

4.2.2 Scatter transform coefficients

A scattering transform is a non-linear signal representation that builds invariance to geometric transformations while preserving a high degree of discriminability. These transforms can be made invariant to translations, rotations (for 2D or 3D signals), frequency shifting (for 1D signals), or changes of scale. These transformations are often irrelevant to many classification and regression tasks, so representing signals using their scattering transform reduces unnecessary variability while capturing structure needed for a given task. This reduced variability simplifies the building of models, especially given small training sets like ours as we have 5 30 seconds' songs per singer.

The scattering transform [13] is defined as a complex-valued convolutional neural network whose filters are fixed to be wavelets and the non-linearity is a complex modulus. Each layer is a wavelet transform, which separates the scales of the incoming signal. The wavelet transform is contractive, and so is the complex modulus, so the whole network is contractive. The result of this transformation gives a reduction of variance and a stability to additive noise of the data. The separation of scales by wavelets also enables stability to deformation of the original signal. These properties make the scattering transform well suited for representing structured signals such as natural images, textures, audio recordings, biomedical signals, or molecular density functions.

In our case, the 1-dimension scattering transform has been used. This transform is the one used with audio signals as they have only one dimension. By computing it, we get three types of coefficients:

- Order 0 coefficients: the average of the signal at the scale 2**(maximum number of scattering scale).

$$S_0 = x \star \phi(t) \qquad \text{(VI)}$$

- Order 1 coefficients: the Mel-frequency cepstral coefficients arranged along time and log-frequency with Mel resolution.

$$S_1 x(t, \lambda_1) = \left| x \star \psi_{\lambda_1} \right| * \phi(t) \qquad \text{(VII)}$$

- Order 2 coefficients: the coefficients along time but with two log-frequency indices: one first-order frequency and one second-order frequency.

$$\left| W_2 \right| \left| x \star \psi_{\lambda_1} \right| = \left(\left| x \star \psi_{\lambda_1} \right| \star \phi, \left| \left| x \star \psi_{\lambda_1} \right| \star \psi_{\lambda_2} \right| \right)_{\lambda_2 \in \Lambda_2} \qquad \text{(VIII)}$$

$$S_2 x(t, \lambda_1, \lambda_2) = \left| \left| x \star \psi_{\lambda_1} \right| \star \psi_{\lambda_2} \right| \star \phi(t)$$

The first and second order coefficients have been computed by setting to 16 the number of wavelets per octave as is used in the instrument recognition task. With this transform provides both Mel-frequency (order 1 coefficients) and modulation features (order 2 coefficients) by recovering averaged lost information.

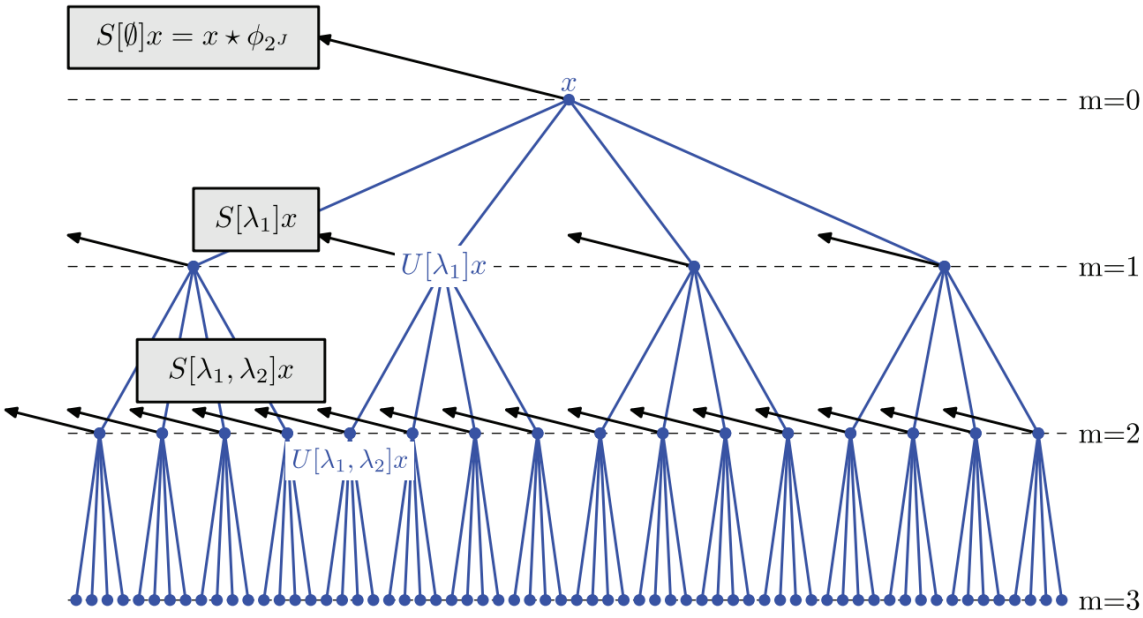

Figure 4-3: Scattering wavelet spectrum [21]

4.3 Classifier

The classification step is based on the implementation of deep learning techniques to classify samples among different classes. This part has been focused from three different points of view: State-of-the-art NNs, SpeakerID networks trained from scratch and SpeakerID networks with pretrained weights.

4.3.1 State-of-the-art Neural Networks

As we have seen, the state-of-the-art NNs are shallow NNs and they are based only on three layers: one input layer, one hidden layer and one output layer.

The input layer is a layer that maps the feature embedding to the hidden layer. It is a universal layer in all of the neural networks and its size depends on the size of the feature embedding of the sample that is processed by the network.

Then, the hidden layer activates with non-linear functions such as sigmoid and soft-max to model the non-linearities of the data. These functions tend to maximize the separability among classes. This makes the clustering of the distribution of the samples that are related to the same singer more accurate:

$$P(t) = \frac{1}{1 + e^{-t}} \qquad (\text{IX})$$

$$P_t(a) = \frac{e^{q_t(a)/\tau}}{\sum_{i=1}^{dim(q_t)} e^{q_t(i)/\tau}} \qquad (\text{X})$$

Finally, in the output layer we find the activation of the neuron related to each singer. In our case, we have needed to insert more neurons in the output layer as we have 200 singers, many more than in the state-of-the-art datasets.

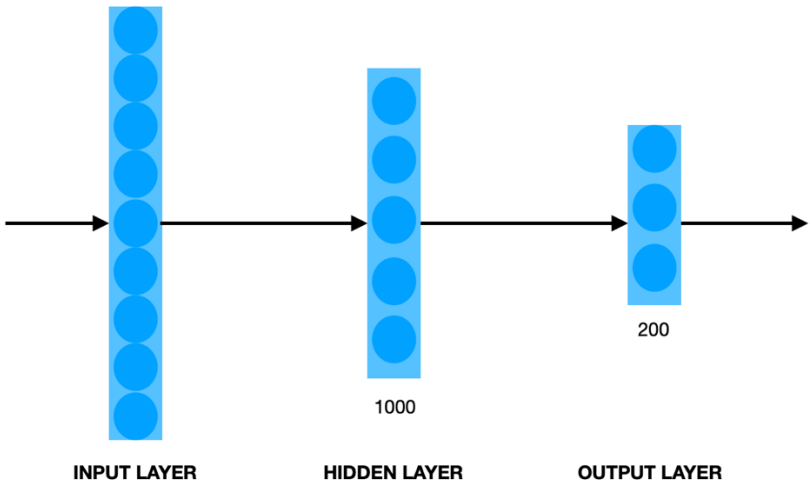

Figure 4-4: State-of-the-art NN adaptation

4.3.2 Speaker ID network trained from scratch

More intricate and sophisticated networks have also been chosen as classifiers. In this section, we have adapted the DeepSpeech network from the audio processing state-of-the-art to work with our problem. This network has been developed to be able to recognize speech with a RNN. The RNN, Recurrent Neural Network, architecture is significantly simpler than traditional speech systems, which rely on laboriously engineered processing pipelines; these traditional systems also tend to perform poorly when used in noisy environments such as our source separated audio files. In addition, this system does not need hand-designed components to model background but instead directly learns a function that is robust to such effects. That is why we have chosen this network. We have trained it from scratch with our dataset to fit the network to work in the Singer identification field.

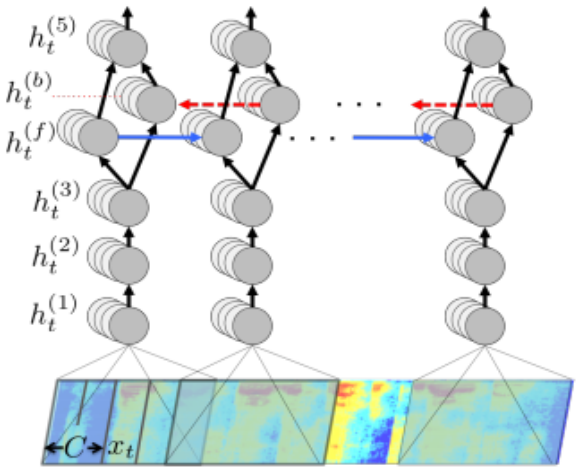

Figure 4-5: DeepSpeech NN [18]

4.3.3 Pre-trained Speaker ID network

As speaker identification is a field with a lot of research done, a state-of-the-art network with pre-trained weights has been adapted to work with our dataset. The biggest advantage that this type of technique has is that the network is already trained and only needs to be adapted to classify our singers. To this purpose, we only the final fully connected layers of the network have to be trained with our train dataset instead of all of the neurons of the network. This fact provides many advantages such as lower computational cost, less execution time and better performance of the network.

From the pre-trained networks of the state-of-the-art, the Speechbrain [17] spkrec-xvect-voxceleb network has been chosen. This is a speaker verification network that works with x-vector embeddings on Voxceleb dataset. This system is composed of a TDNN model coupled with statistical pooling and is trained with Categorical Cross-Entropy Loss. The data that has been used to train this network is the Voxceleb dataset. Therefore, as we want to identify singers a part of which are contained in the Voxceleb dataset, we expect to have better performance results than the other types of classification networks.

5 Integration and experimental results

Many experiments have been developed in order to be able to compare the performance of the system changing the audio features used as embedding of the neural networks and the classifiers that perform the singer estimations. The programming language that has been chosen is Pytorch as it is one of the most spread programming languages in the field of deep learning. Next, the sequence of experiments carried out is explained.

5.1 State-of-the-art NN with x-vector as features

The first system that has been build is the one containing the networks from the state-of-the-art as classifiers. Due to the high computational complexity of the system, the code has needed to be re-arranged three times in order to make it as light as possible in terms of computational cost and RAM utilization.

A shallow NN has been implemented using a hidden layer of 1000 neurons as the ones in the state-of-the-art, sizing its dimensionality to be higher than the number of neurons in the output layer. Moreover, from the feature extraction point of view, Facebook's wav2vec2.0 has been implemented to extract the x-vector feature embeddings of each audio file. In image 5-1, the Cross-Entropy loss of the system can be seen. These graphics show the performance of the system with Adam optimizer and Cross Entropy loss function with learning rate of 0.01 and 100 epochs. The scheduler that manages the variation of the learning rate has been studied to begin to decrease the learning rate value at epoch 20 and at epoch 50 as can be seen in the image 5-1 without having significant results to our system's performance. In addition, the batch size of samples that are feeding to the network has been set to 200 to make the execution of the code faster.

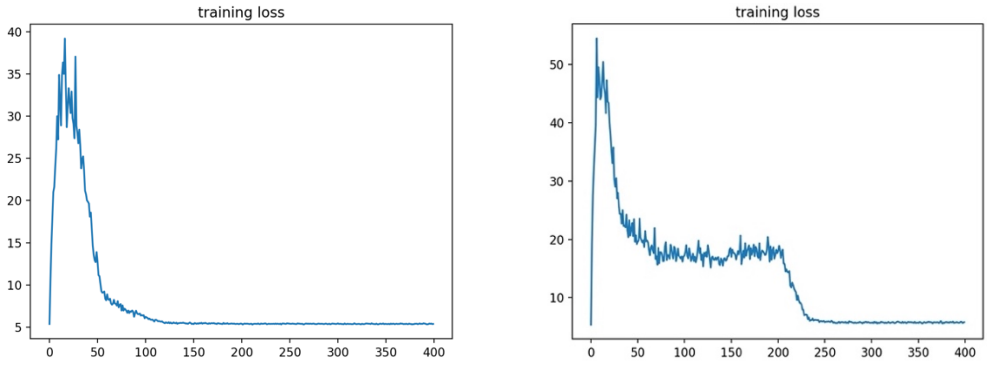

Figure 5-1: X-vectors + SoA NN error performance

As can be seen in the figure above, the minimum loss that has been achieved is around 5 points, which is a pretty high value for a reliable classification network. In addition, the number of singers that the network has been able to identify has been 13 out of 200. These results should be improved to build a reliable singer identification system.

5.2 State-of-the-art NN with scatter transform coefficients as features

As the previous experiment has not performed accurately with our data, we have changed the features extracted from the audio signal in order to model better the singing voice. For this purpose, the scattering transformation of the signal has been computed. This transformation is used to identify instruments and sounds as it describes the frequency of the signal in an accurate way. Its problem is that if a detailed representation of the signal is needed to be obtained, it needs a lot of coefficients to describe the signal.

After analyzing the coefficients of the signal obtained from the application of the transformation to our data, the signal obtained has been processed. In figure 5-2 the order 0 coefficients of a song from the database is shown. As they only provide an average of the signal in time, they have been discarded. This utility is not useful to analyze the frequency characteristics of the signal. In figure 5-3 the scatter coefficients of order 1 a signal of our dataset are plotted. As the frequency scale is the Mel-frequency scale, they are the Mel-Frequency Cepstral Coefficients of the signal. In addition, the more detailed order 2 coefficients of the scatter transform are shown in figure 5-4. These coefficients show more detailed information about the modulation of the frequency of the signal as can be seen in the image.

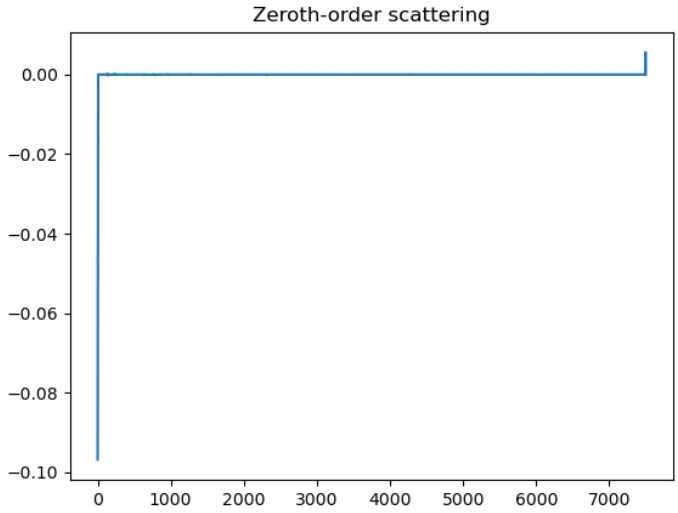

Figure 5-2: Scatter transform coefficients of order 0 of a song from the dataset

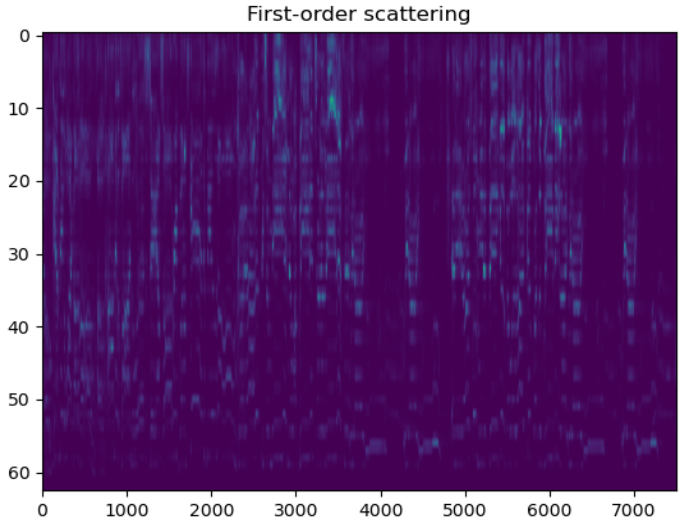

Figure 5-3: Scatter transform coefficients of order 1 of a song from the dataset

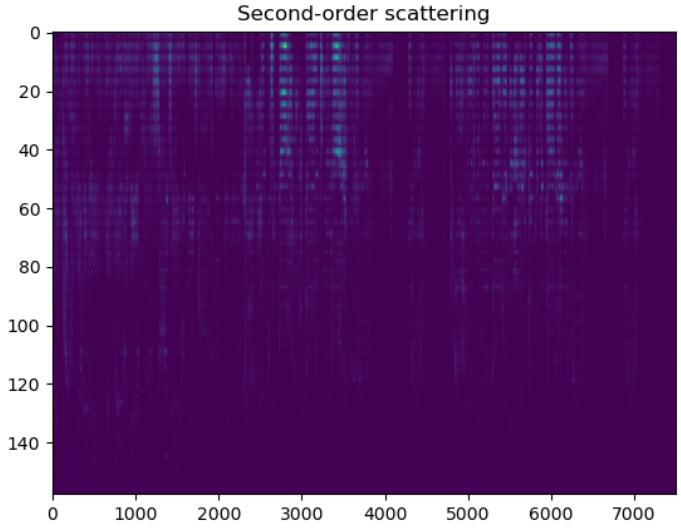

Figure 5-4: Scatter transform coefficients of order 2 of a song from the dataset

Therefore, the network has been trained with the scatter coefficients that carry frequency information of the song. This fact has decreased the computational cost significantly. In figure 5-5 the performance of the network is shown. The minimum loss that has been obtained is of 5.2 points, which is still a high value for a reliable classification system. As from the accuracy point of view, it has been increased reaching the 24 out of 200, which is still a low value.

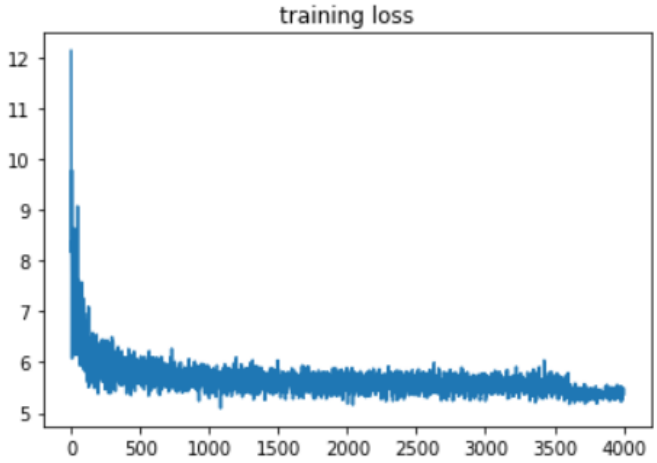

Figure 5-5: Scatter coefficients + SoA NN error performance

5.3 State-of-the-art NN with x-vector and scatter transform coefficients as features

The following investigation path that has been tested is to use a more complex representation of the signal. To achieve this, we have concatenated the x-vectors obtained from the wav2vec2.0 extractor and the scatter transform coefficients previously computed.

The performance of the system provides a loss of 1.4 points, which is a better result than with the other features but the accuracy of the system has not been increased from the previous experiments' accuracy. Therefore, this network does not identify the singers better than the previous one although it has a lower cross-entropy loss than the other experiments.

After doing these three experiments, their performance has been compared to the state-of-the-art ones. The state-of-the-art accuracy is around the 90%, which has been demonstrated to be impossible to achieve with this simple network. So, it is clear that the state-of-the-art experiments are over-fitted to the data that they have used, being these biased and specific datasets instead of generic datasets unbiased to musical genre and song style as is our dataset. By this, we can conclude that the state-of-the-art metrics are not realistic values for the metrics used to analyze the performances of the networks.

5.4 Speaker recognition network from scratch with x-vector as features

There are some networks available in the state-of-the-art of speech processing. The Deep Speech network from the torchaudio repository [15] is the one that has been selected to be used in our system. This speech neural network has been modified to estimate our 200 singers out of our dataset. The performance of this experiment took much more time than the

previous experiments. This fact is due to the increment in computation cost derived from training not only three layers but also many more like this system has.

As it is a big network and we do not have a large dataset in terms of hours of audio files, the metrics obtained are not competitive. The minimum loss obtained is of 47.3 points and the system can only identify eight singers out of the 200 from the dataset. This big loss result and the scarce singers' identification accuracy states that more data will be needed to improve the performance of the system.

5.5 Speaker recognition network from scratch with scatter transform coefficients as features

As the previous experiment has not obtained good results, the features extracted by the model have been changed to the scatter transform coefficients. This change has made the execution of the experiment impossible due to its extremely high computational cost. Its main problem is the necessity of a RAM with more capacity than the available from the machines used.

5.6 Speaker recognition network from scratch with x-vector and scatter transform coefficients as features

Like it has been seen in previous experiments, the election of a model centered in the frequency of the signal provides better results in the performance of the classification systems but the combination with the x-vectors is supposed to achieve better results in terms of performance metrics. Therefore, we have changed the previous scatter transform coefficients to a concatenation of the x-vectors and the scatter transform coefficients. In the case of this experiment, the implementation of the system has been carried out but it has also been impossible to analyze the performance of the system. The performance of the system with these features has not been able to be analyzed due to the high RAM memory necessity.

5.7 Pre-trained speaker recognition network from scratch with x-vector as features

The last experiment that has been developed is the one with the speaker identification pre-trained network. For this purpose, the SpeechBrain pre-trained TDNN network with x-vectors has been chosen. In order to work with our database, the network has been tried to be fine-tuned and its last layers trained but, due to its excessively high computational cost, this option has been modified. This has made us possible to do the experiments without saturating the RAM memory of the computing machines. As this is an already trained network used for speaker recognition, its performance is supposed to be better than the other

neural networks. So, the performance of the network has been analyzed in inference mode with the weights related to the Voxceleb dataset.

This experiment is based on the idea of transfer learning. With transfer learning, the networks are able to classify a type of data although they have been trained with another type of data. In our case we want to see if a network that is adapted to identify speakers and has been trained with a large speech dataset can also identify singing voice. In order to evaluate the quality of the performance of the network, the confusion matrix of the predictions has been computed. As an example, the confusion matrix obtained from one songs of the dataset is shown in figure 5-6. This image shows that the network is able to identify with a high precision the samples that are not related to a singer but is not able to identify the 5 songs related to the singer.

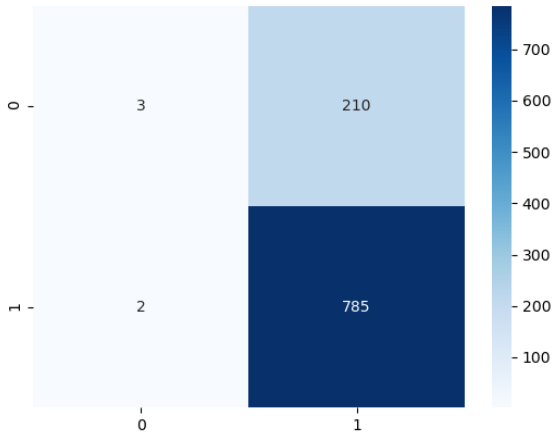

Figure 5-6: Confusion Matrix of a song of the dataset

In figure 5-7 the performance of the first songs of Pop, Rock and Jazz musical genres are shown. It can be concluded from the figure that each song behaves in a different way when it is processed by the system. Nevertheless, the accuracy of all of the songs is greater than 0.684 which means that the system behaviors is noticeably good independently to the musical genre being processed.

Musical Genre	Artist	Song	Song ID	metrics	accuracy	coincidences
POP	King Princess	House Burn Down	48oBIuYRNSsKPigKQAQzXp	TP: 3, FP: 210, FN: 2, TN: 785	0.788	['King Princess', 'King Princess', 'King Princ...
ROCK	Marmozets	Meant To Be	4XII7kI1yxHf4DDDFf2CYu	TP: 5, FP: 316, FN: 0, TN: 679	0.684	['Marmozets', 'Marmozets', 'Marmozets', 'Marmo...
JAZZ	Rebecca Ferguson	The Wave - Rogerseventytwo Remix	6L2ig70QNCKbT872NcoFJs	TP: 2, FP: 175, FN: 3, TN: 820	0.822	['Rebecca Ferguson', 'Rebecca Ferguson']

Figure 5-7: Comparison of performance of the system in one song of each genre

To sum up all the results, table 1 has been built with the most significant details of each of the experiments.

In this table, the train and test accuracy of each experiment done can be found. As it can be seen, the accuracies of the networks that have been trained from scratch are too low to be able to identify correctly very few singers from their songs. Instead, the accuracy obtained performing the inference in the experiment with the pre-trained weights is shows that the transfer learning technique increases the quality of the performance of the system. Having a 0.721 accuracy makes this the most reliable of the systems that have been tested.

This statement has also been supported by the cross-entropy loss value obtained from analyzing the performances of the experiments. The pre-trained experiment is the one that has been able to reach a cross-entropy loss value closer to 0, which is the best value.

The last metric included in the table is the computation cost. This is expressed as: medium cost, if it has taken less than two hours and less than 10 GB of RAM; high cost, if it has taken between two and five hours and 10-20 GB of RAM; very high cost, if it has taken more than 24 hours and 20-35 GB of RAM; and extreme cost, if it has not been able to be executed due to its need of more than 48 hours and more than 35 GB of RAM to be executed. From this metric it can be concluded that these experiments need high quality computation systems to be done which makes the experiments 5 and 6 unaffordable for the systems available.

The optimization of the parameters of the systems has not been able to be done due to the high computational cost of the experiments although it is an interesting line to be followed as future work to increase the quality of the performance of the systems to their maximum level.

EXPERIMENT	TRAIN ACCURACY	TEST ACCURACY	CROSS-ENTROPY LOSS TRAIN	CROSS-ENTROPY LOSS TEST	COMPUTATION COST
1	0.1225	0.065	5.4	6.3	MEDIUM
2	0.1888	0.12	5.2	5.9	MEDIUM
3	0.1537	0.1	1.4	1.5	HIGH
4	0.0837	0.04	47.3	103.23	VERY HIGH
5	—	—	—	—	EXTREME
6	—	—	—	—	EXTREME
7	—	0.721	—	0.13	VERY HIGH

Table 2. System performance

6 Conclusions and future work

6.1 Conclusions

As a conclusion, this is a really challenging problem with a lot of difficulties that have to be furthered studied.

The first problem that has been solved practically perfect is the lack of datasets in the state-of-the-art to work with. Our proposal has been a universal dataset created using the Spotify's API. With this new dataset, research regarding music and singing voices can be done. Moreover, a batch of song's features have been collected to be able to analyze the performance of the systems developed using this data, not only from the variation of artists point of view, but also from musical features such as the musical genre or the loudness of the song. Therefore, our contribution to the singer identification state-of-the-art is supposed to be useful for many researchers as a high-quality dataset is the base of any deep learning system.

Nevertheless, some improvements can be done to our method proposal for singer classification. The state-of-the-art networks of singer identification have been demonstrated to be inefficient when we increase the number of singers to be identified and the variety of musical genres and songs. The complexity and amount of the data have made impossible to carry out some of the experiments with the computational resources that were available. For this purpose, further experiments with better machines are needed to be carried out in order to fully analyze our feature proposal model.

6.2 Future work

As future work, it has been concluded that there are two ways that can be studied to improve the performance of our system which are: the improvement of the dataset and the improvement of the system developed used to distinguish among the singers.

1. System improvements:
    a. Adaptation of other speaker id network models to work with our audio features.
    b. Run the experiments in better computational machines.
    c. Apply different Transformers' architectures to extract more potential features.
    d. Optimize the parameters of the networks
2. Dataset improvements:
    a. Enlarge the database incorporating male singers with their respecting song audio files in order to build a system invariant to the singer's gender.

b. Change the audio files obtained from downloading the Spotify's preview songs to the complete audio songs files obtained from YouTube API.

c. Compare the performance between the source separation carried out by the Deezer's Spleeter application and a latest model in Music source separation carried out by Facebook which is DEMUCS [16] application.

Bibliography

[1] W. Tsai and H. Lee, "Singer Identification Based on Spoken Data in Voice Characterization," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 8, pp. 2291-2300, Oct. 2012, doi: 10.1109/TASL.2012.2201473. https://ieeexplorer.ieee.org/document/6205358

[2] W. Cai, Q. Li and X. Guan, "Automatic singer identification based on auditory features," 2011 Seventh International Conference on Natural Computation, Shanghai, China, 2011, pp. 1624-1628, doi: 10.1109/ICNC.2011.6022500., https://ieeexplore.ieee.org/document/6022500

[3] Mesaros, Annamaria & Virtanen, Tuomas & Klapuri, Anssi. (2007). "Singer Identification in Polyphonic Music Using Vocal Separation and Pattern Recognition Methods." 375-378. https://www.researchgate.net/publication/220723246_Singer_Identification_in_Polyphonic_Music_Using_Vocal_Separation_and_Pattern_Recognition_Methods

[4] Chin-Chin Liu and Chuan-Sung Huang. 2002. "A singer identification technique for content-based classification of MP3 music objects". In Proceedings of the eleventh international conference on Information and knowledge management (CIKM '02). Association for Computing Machinery, New York, NY, USA, 438-445. https://doi.org/10.1145/584792.584864

[5] Ratanpara, Tusharkumar & Patel, Narendra. (2015). "Singer identification using perceptual features and cepstral coefficients of an audio signal from Indian video songs." EURASIP Journal on Audio, Speech, and Music Processing. 2015. 10.1186/s13636-015-0062-9.

[6] Kim, Youngmoo & Whitman, Brian. (2002). "Singer Identification in Popular Music Recordings Using Voice Coding Features." https://www.researchgate.net/publication/2563406_Singer_Identification_in_Popular_Music_Recordings_Using_Voice_Coding_Features

[7] Biswas S., Solanki S.S. (2019). "Singer Identification Based on Artificial Neural Network." In: Luhach A., Jat D., Hawari K., Gao XZ., Lingras P. (eds) Advanced Informatics for Computing Research. ICAICR 2019. Communications in Computer and Information Science, vol 1075. Springer, Singapore. https://doi.org/10.1007/978-981-15-0108-1_36

[8] https://developer.spotify.com/documentation/web-api/

[9] David Martín-Gutiérrez, SpotMux: A data collector for Spotify & MusixMatch, (2020), GitHub repository, https://github.com/dmgutierrez/spotify-musixmatch-data-collector#spotmux-a-data-collector-for-spotify—musixmatch

[10] Andrew Jiang, Wav Splitter API (aka Splitterkit), (2016), GitHub repository, https://github.com/abvthecity/wav-splitter-library

[11] Avinash Narayanan, Singer Identifier, (2020), GitHub repository, https://github.com/avi-narayanan/singer-identifier

[12] Dabike, Gerardo Roa and J. Barker. "The Use of Voice Source Features for Sung Speech Recognition." ArXiv abs/2102.10376 (2021): n. pag.

[13] Andreux M., Angles T., Exarchakis G., Leonarduzzi R., Rochette G., Thiry L., Zarka J., Mallat S., Andén J., Belilovsky E., Bruna J., Lostanlen V., Hirn M. J., Oyallon E., Zhang S., Cella C., Eickenberg M. (2019). “Kymatio: Scattering Transforms in Python.” arXiv preprint arXiv: 1812.11214.

[14] Baevski, Alexei, Henry Zhou, Abdel-rahman Mohamed and Michael Auli. “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.” *ArXiv* abs/2006.11477 (2020): n. pag.

[15] https://pytorch.org/audio/stable/models.html#hannun2014deep

[16] Défossez, Alexandre, Nicolas, Usunier, Léon, Bottou, and Francis, Bach. "Music Source Separation in the Waveform Domain". arXiv preprint arXiv:1911.13254 (2019).

[17] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. "SpeechBrain: A General-Purpose Speech Toolkit." (2021).

[18] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. "Deep Speech: Scaling up end-to-end speech recognition." (2014).

[19] Snyder, David, Daniel, Garcia-Romero, Daniel, Povey, and Sanjeev, Khudanpur. "Deep Neural Network Embeddings for Text-Independent Speaker Verification.”. Interspeech 2017 (pp. 999-1003).2017.

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5329-5333, doi: 10.1109/ICASSP.2018.8461375.

[21]. J. Anden and S. Mallat. Deep scattering spectrum. IEEE Transactions on Signal Processing, 62(16):4114–4128, 2014.

[22] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. "Spleeter: a fast and efficient music source separation tool with pre-trained models”. Journal *of Open Source Software* 5, no.50 (2020): 2154.

Glossary

API	Application Programming Interface
MIR	Music Information Retrieval
MFCC	Mel-Frequency Cepstral Coefficients
LPC	Linear Predictive Coding
WLPC	Warped Linear Predictive Coding
LPMCC	Linear Prediction Mel-Frequency Cepstral Coefficients
GTCC	Gamma Tone Cepstral Coefficients
DCT	Discrete Cosine Transform
MDCT	Modified Discrete Cosine Transform
GMM	Gaussian Mixture Model
K-NN	K – Nearest Neighbors
EM	Expectation Maximization
KL	Kullback – Leibler
SVM	Support Vector Machine
PCA	Principal Component Analysis
LM	Levenberg - Marquard
BR	Bayesian Regularization
SCG	Scaled Conjugate Gradient
OSS	One-Step Secant back-propagation
NN	Neural Network
TDNN	Time Delay Neural Network
RNN	Recurrent Neural Network