Emotion from music spectrograms

Matteo Cerutti
Politecnico di Torino
s265476@studenti.polito.it

Antonio Santoro
Politecnico di Torino
s264014@studenti.polito.it

Marco Testa
Politecnico di Torino
s265861@studenti.polito.it

Abstract

1. Introduction

Nowadays people need to have the possibility to select music and make playlists based on their mood. Many music platforms feature different music playlists made by hand that include popular and commercial songs aiming to maximise ratings. One of the most used feature on these platforms is to create playlists similar to other ones, the point is that all the songs that will be included are selected on the "similarity".

After some researches, we found that our idea was applied to classify song genres, therefore starting from the article of Piotr Kozakowski and Bartosz Michalak [2], we adapted their work to our objective.

The interest is to train a neural network on different audio speeches that represent different human emotions, extract features and try to see whatever those peculiarities can be matched from music. Amiriparian *et al*. [1] showed that processing spectrograms into networks characterized by a different depth the result will change. This report presents results obtained from three networks, ResNet, VGG and GoogLeNet, trained on the RAVDESS Emotional song audio dataset [4] and tested on the CAL500 dataset [5].

This work is organized as follows. In Section 2 we give an explanation on how the training and testing dataset have been preprocessed to be adapted to our purposes, how the samples have been filtered and selected. Section 3 focuses on the networks training phase in which useful hyperparameters sets have been chosen in order to obtain significant validation and training results. In Section 4 we present the testing algorithm that classifies each song. Finally, some conclusions and suggestions for future work are drawn in Section 5.

1.1. Classification pipeline

A simple image classifier could have submitted poor performances, thus let the network training compatible with the variable length of each song, we figured out a model that is capable of slicing each song, treated as variable sized spectrograms as well, then choose a label after checking the rank of each slice that compose the whole track. Figure 1 shows the general model structure, in the final step the song will be classified by means of a voting algorithm.

2. Data preparation

2.1. Training dataset

The RAVDESS Emotional song audio consists of 1012 files of actors singing in a neutral North American accent. The portion used for this work includes calm, happy, sad, angry, and fearful emotions, each vocal is produced at two levels of emotional intensity, normal and strong.

Files are provided as .wav (16bit, 48kHz, mono, 4 seconds each) that need to be converted into raw spectrograms. For the purpose "SoX (Sound eXchange) sound processing utilities" has been used. This tool can process audio files and do things like trimming or filters frequencies. Spectrograms for the training dataset have been generated to fit the input size of the three networks, furthermore, to cope with the limited size of the dataset, augmentation has been applied like random grey colorizing, brightness, contrast and hue variations. Amiriparian *et al*. [1] showed that using different shade of colour could exhibit different outcomes. Unfortunately, since the hue transformation made by the PyTorch framework has a not negligible impact on the brightness of the image and the goal is to make any alteration on the information of the spectrograms, the best choice was to stick with the original shades (Figure 2), hence playing with the contrast and with monochromes image could have a positive effect on capturing some features.

2.2. Test dataset

The CAL500 dataset contains 500 songs performed by 500 unique artists, each song has been annotated by at least three people using a standard survey. Files are provided as .mp3 (32kbps, mono) along with one or more labels.

As Liu *et al*. [3] stated, "preprocessing the spectrograms is a key point of successfully applying CNN on music spectrograms", because the input to CNN requires to be a fixed

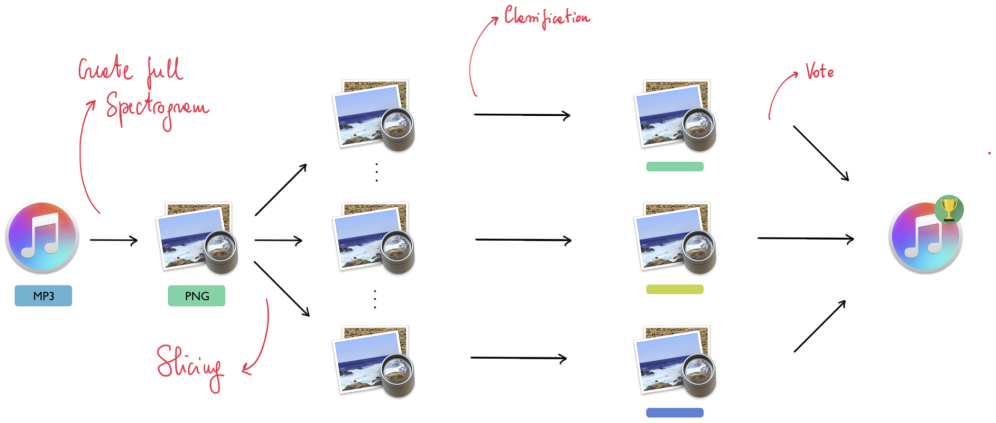

Figure 1. Classification pipeline.

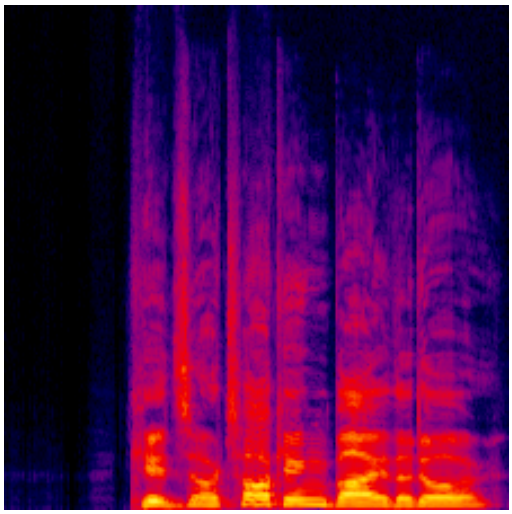

Figure 2. Training spectrogram sample.

size matrix and each track has a variable duration. The point is to let the network be able to recognize the emotion by taking a closer look to the song. The simplest approach would have been to extract spectrograms of the desired size after computing the number of overlaps. Yet, the chosen approach has been to not loose any data during the preprocessing step, thus defining an algorithm that will generate spectrograms with variable length in order to match it to the duration of the tracks. In order to make the dataset compatible with our testing environment, two actions have been performed.

2.2.1 Filtering

Since the training label set was a subset of the CAL500 labels, we selected only the songs which classes belong to the first set. Furthermore, an additional filtering step has been performed to remove all the redundant classes keeping only the relevant ones.

2.2.2 Slicing

The most challenging step was making the test dataset compatible with the training samples. To cope with the variable duration of each song, the extracted spectrograms have been sliced into squared images to fit to the network input size without losing any information. Each slice has been generated by sampling a proportional quantity of information equal to the training samples of four seconds. In fact, SoX allows to extract the spectrogram by setting the number of pixel per second and the input size of the image. Given that, a different testing approach has been implemented.

3. Training phase

The research method to find the optimal set of hyperparameters has been the same for all networks[1] except for a slight difference related to GoogLeNet due to its three output branches. A lot of experiments have been done on different network variants of the same model to evaluate the impact of the networks' depth on the results.

The first step was to find a good starting hyperparameters set to make the network accomplish a full training. Due to Google Colab limitations, a random search has been used to

[1]ResNet50 & ResNet152, VGG11 & VGG19 and GoogLeNet (Inception v1)

Network	LR	BS	WD	G
ResNet152	0.003	12	2e-05	0.6
ResNet50	toadd	toadd	toadd	toadd
VGG19	0.003	8	3e-04	0.01
VGG11	0.0005	8	3e-05	0.05
GoogLeNet	0.0001	8	5e-05	0.1

Table 1. Best values per hyperparameter.

Network	Validation accuracy
ResNet152	50%
ResNet50	40%
VGG19	50%
VGG11	65%
GoogLeNet	82%

Table 2. Average validation accuracy per network in 100 epochs.

evaluate 50 different hyperparameters sets, the approximate best ones have been reported in Table 1.

Using the reported sets, all the networks have been trained for 100 epochs and evaluated using different split ratios between training set and validation set. The final values have been selected by doing some tuning by hand after evaluating the networks' perfomance during the epochs, moreover the values of each hyperparameter have been adjusted to address the problem of the high epochs number and to prevent the occurring overfit. Since the training dataset is very small, we dealt with the overfit problem by means of data augmentation, yet no significant improving has been noticed.

Table 2 contains the average best validation scores calculated on the validation set per network and variants. Due to lack of time, different combinations of criterions and optimizers have not been tested, however we observed that increasing that decreasing the size of the validation set the overall score of a network similar to GoogLeNet is decent. Although the training dataset is small, applying a significative amount of regularization prevented the occurring of the overfitting phenomenon seen right after half of the epochs, but the limitations imposed by the platform kept us from increasing the number of epochs and evaluted more sets of hyperparameters.

4. Testing

5. Conclusions

This report proposed a method to classify songs with emotions captured from human vocal singing recordings.

References

[1] Shahin Amiriparian, Maurice Gerczuk, Sandra Ottl, Nicholas Cummins, Michael Freitag, Sergey Pugachevskiy, Alice Baird, and Bjorn Schuller. Snore sound classification using image-based deep spectrum features. 2017.

[2] Piotr Kozakowski and Bartosz Michalak. Music genre recognition. 2016. `http://deepsound.io/music_genre_recognition.html`.

[3] Xin Liu, Qingcai Chen, Xiangping Wu, Yan Liu, and Yang Liu. Cnn based music emotion classification. 2017.

[4] Livingstone SR and Russo FA. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. plos one 13(5): e0196391, 2018. `https://doi.org/10.1371/journal.pone.0196391`.

[5] Turnbull, Douglas, Barrington, Luke, Torres, David, Lanckriet, and Gert. Semantic annotation and retrieval of music and sound effects. *Audio, Speech and Language Processing, IEEE Transactions on*, 16(2):467–476, 2008.

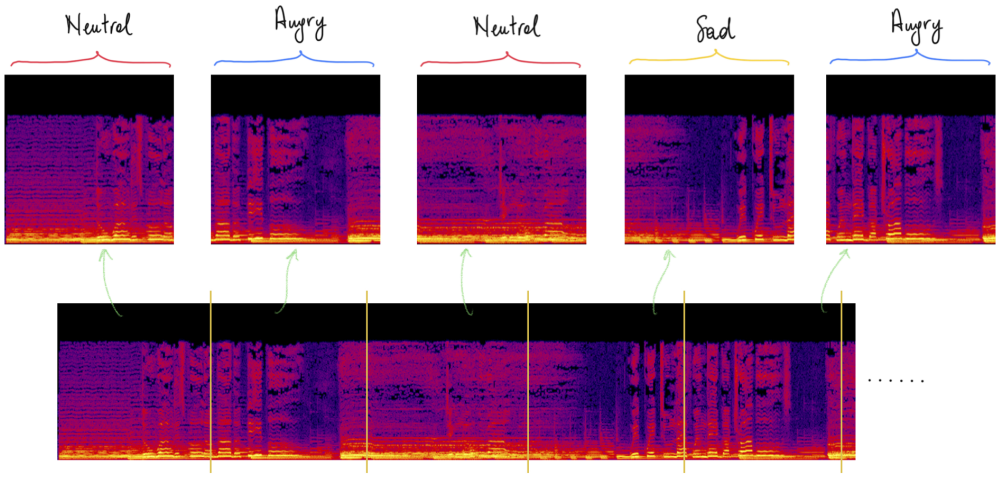

Figure 3. Voting system.