Song Emotion Recognition: a Performance Comparison Between Audio Features and Artificial Neural Networks

Pedro Benevenuto Valadares[1], **Karen Gissell Rosero Jácome**[1], **Arthur Nicholas dos Santos**[1], **Bruno Sanches Masiero**[1]

[1]Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Av. Albert Einstein, Nº 400 - Cidade Universitária,
Campinas, Brazil - SP, 13083-852

{p204483@dac., k264373@g., a264372@dac., masiero@} unicamp.br

Abstract. *When songs are composed or performed, there is often an intent by the singer/songwriter of expressing feelings or emotions through it. For humans, matching the emotiveness in a musical composition or performance with the subjective perceptiveness of an audience can be quite challenging. Fortunately, the machine learning approach for this problem is simpler. Usually, it takes a data-set, from which audio features are extracted to present this information to a data-driven model, that will, in turn, train predicting the highest probability of an input song matching a target emotion. In this paper*[†]*, we studied the most common features and models used in recent publications to tackle this problem, revealing which ones are best suited for songs a cappella.*

1. Introduction

Music is art, and art is a form of expression. By the time it reaches an audience, a spectrum of emotional reactions can be provoked, in account that music emotion recognition (MER) is a process that is highly intertwined with people's life experiences and cognitive capacities. In contrast, as a sub-field of music information retrieval (MIR), MER deals with classification of music according to affective computing [Kim et al. 2010].

Currently, the importance of MER can be justified by the dependency of search and recommendation engines on metadata, which, in simple terms, is just data about data. When a person uses a smartphone to take a picture of a cat, for instance, the data is the picture itself. However, a series of other information about that data is also recorded, e.g., the time, date, and geographical coordinates where it was shot etc. All of which are metadata that can be used to tag the data itself, to either retrieve it in the future or find similar content. Likewise, song metadata, e.g., genre, composer, artist, album, year of release etc., are commonly used by streaming services to help users find what they may be prone to like, or even recommend songs based on their listening history. However, the mood of a song is also an interesting metadata, that could be used to relate a certain song to similar content.

[†]This work was partially supported by the São Paulo Research Foundation (FAPESP), grants #2019/22795-1 and #2019/22945-3. The opinions, hypothesis and conclusions or recommendations expressed in this material are the authors' responsibilities, and not necessarily reflect FAPESP's views.

In this paper, we studied various articles with *state-of-the-art* results published on MER, for both song and instrumental music. Our findings reveal that, usually, timbral features, e.g., Mel spectrogram, Mel-frequency *cepstral* coefficients (MFCC) etc., are employed as front-ends, whilst regarding artificial neural network (ANN) models, the multi-layer perceptron (MLP) and convolutional neural network (CNN) architectures are most commonly employed as back-end. However, when comparing the performance of different audio features and ANN models, our experiments showed that the chromagram, which is a harmonic feature, combined with either one-dimensional (1-D) or two-dimensional (2-D) CNN architectures yields even better results.

The remainder of this paper is organized as follows: Section 2 details our findings on what are the most commonly used audio features to represent music samples to the most commonly used machine learning (ML) model architectures. Section 3 describes our experiment, in which different ANN models were synthesized based on the information retrieved from the previous section. Section 4 shows our results, comparing it to previous *state-of-the-art* works that used a same data-set as we did. Finally, Section 5 presents some pertinent considerations to conclude this study.

2. Features and Models

According to [Panda et al. 2020], musical dimensions can be related to emotions by a set of high-level features, namely: melody, harmony, rhythm, dynamics, tone color (timbre), expressivity, texture, form, and vocals. On the other hand, computational features are considered low-level, because they only provide primitive descriptions by which individual high-level ones may be identified.

2.1. Audio features

By reviewing the works of [Flamia Azevedo and Bressan 2018], [Gao et al. 2020], [Casper 2020], [Russo et al. 2020], [Kim 2020], [Pandrea et al. 2020], [Du et al. 2020], [Ospitia Medina et al. 2020], [Rajesh and Nalini 2020] and [Cunningham et al. 2020], we found 47 different low-level computational features being used separately or concatenated, to better represent training data-sets, depending on different ML architectures used. All these features are available *off-the-shelf* on Python libraries and MATLAB toolboxes, and 6 of them were found to be used on 76.6% of the publications reviewed (each):

- *Chromagram*: relates to harmony, i.e., the sound produced by the combination of various pitches, and indicates energy distribution along a 12-dimensional vector (one for each semitone in the super-just harmonic scale, i.e., from A to G#.);
- *Mel spectrogram*: relates to tone color, i.e., timbre, and decomposes an audio signal into a series of frequency channels inspired by the human cochlea, enabling to study the signal's frequency distribution into so-called critical bands;
- *Mel-frequency cepstral coefficients (MFCC)*: also relates to tone color and measures spectral shape. Can be derived from a log magnitude Mel spectrogram based on the discrete cosine transform (DCT). Typically, only the first 8 to 13 MFCCs are used for voiced signals;
- *Spectral centroid*: also relates to tone color and represents the mean of the magnitude spectrum of the short-time Fourier transform (STFT);
- *Spectral roll-off*: also relates to tone color and indicates the frequency below which approximately 85% of the magnitude spectrum distribution is concentrated;

- *Zero-crossing rate (ZCR)*: also relates to tone color and represents the number of times a waveform changes sign in a window, indicating change of frequency and noisiness.

As for the other 41 audio features (which were used on the other 23.4% of the publications reviewed), 13 of them were related to rhythm, 10 were related to tone color, 6 were related to harmony, 5 were related to melody and dynamics (each), and only 1 was related to texture, musical form and vocals (each).

2.2. ML models

According to [Flamia Azevedo and Bressan 2018], what dictates which and how many features can be used as front-end for an ML model is the architecture of the model itself. By reviewing the works of [Flamia Azevedo and Bressan 2018], [Gao et al. 2020], [Casper 2020], [Russo et al. 2020], [Kim 2020], [Pandrea et al. 2020], [Du et al. 2020], [Ospitia Medina et al. 2020], [Rajesh and Nalini 2020] and [Cunningham et al. 2020], we found 12 different architectures being used, separately or combined. 3 of these were found to be used on 17% of the publications reviewed (each), namely: support vector machine (SVM), multi-layer perceptron (MLP), and convolutional neural network (CNN); and another 2 on 10% of the publications reviewed (each), namely: recurrent neural network (RNN) with long short-term memory (LSTM) blocks, and random forest. Since the initial purpose of our work was to compare the performance of ANN models, we opted to leave out SVM and random forest, which are ML models that are not based on ANNs, thus focusing on the remaining 3 models:

- *MLP*: is a type of ANN that models the relationship between a set of training data and a group of known targets. Its architecture is based on a simplified understanding of how the human brain responds to stimuli from sensory organs and is best suited to problems where the relationship between input and output data is well understood, yet the process that relates both is extremely complex;
- *CNN*: is a type of ANN based on convolutional operations that can extract high-level features from 1-D or 2-D low-level ones. It can deeply extract underlying features contained in each frame, while retaining time-series features in the same direction. In classification problems, an MLP layer is usually employed at the end of a CNN architecture, to output its predictions;
- *RNN-LSTM*: is a type of ANN which commonly relies on sequential data, i.e., time-series vectors. In the special case of an LSTM block, it can learn dependencies in the time-dimension of time-frequency (T-F) features, incorporating local context in ANN predictions. Hence, when trained on top of CNNs, for instance, the input data is no longer an individual *"image"*, e.g., a spectrogram, but rather a sequence, more like in a *"movie"*. Moreover, when using a bidirectional LSTM (Bi-LSTM) block, it can also handle the forward and backward flow of information in ANNs.

As for the other 7 architectures, k-nearest neighbors (K-NN) was found to be used on 7% of the publications reviewed, RNN with gated recurrent unit (GRU) blocks, decision tree (CART and C4.5) and state vector regressor (SRV) were found to be used on 4% of the publications reviewed (each) and logistic regression was found to be used on 3% of the publications reviewed.

3. Experimental setup

To experiment with the aforementioned features and models, a portion of the Ryerson Audio-Visual Database of Emotional Speech and Song (`RAVDESS`) data-set was chosen, which comprises $1,012$ audio-only files of song recordings, performed by 24 actors, singing 2 lexically matched statements, in a neutral North American accent. Song emotions include neutral, calm, happy, sad, angry, and fearful expressions [Livingstone and Russo 2018]. To extract audio features from this data-set, the Python package for music and audio analysis `Librosa` (version $0.8.0$) was used, and all ANN models were synthesized using `TensorFlow` (version $2.2.0$), compiled using ADAM optimization, *categorical cross-entropy* as loss function, and trained using an NVIDIA TITAN V graphics processing unit (GPU).

3.1. MLP model

Since an MLP model can have an input layer with as many neurons as necessary, all input features could be concatenated and flattened into an 1-D input vector. After extracting all 6 audio features from the data-set, the principal component analysis (PCA) technique was used, to visualize the minimum number of variables that keeps the maximum amount of information about how each feature data is distributed, as illustrated in Figure 1.

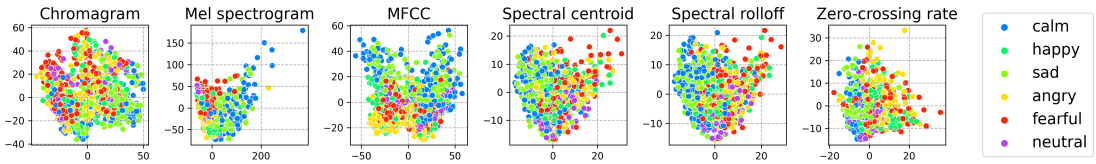

Figura 1. PCA plots for all 6 audio features.

Since Mel spectrogram showed the worst clustering and the highest dimensionality compared to the other features, we chose to train an MLP model using the concatenation of the other 5, i.e., chromagram, MFCC, spectral centroid, spectral roll-off and ZCR. Hence, our MLP model consisted of an input layer with $11,394$ neurons, followed by 2 hidden layers with $1,024$ and 128 neurons each, totaling over 141M free parameters. Rectified Linear Unit (ReLU) was used as an activation function for all layers, except for the output one, where *softmax* was used. For regularization, we used *dropout*, which randomly ignores a percentage of neurons during training.

3.2. 2-D CNN model

For our 2-D CNN model, we selected the 2-D feature that showed the clearest clustering in the PCA visualization in Figure 1, i.e., the chromagram. Thus, the input for this architecture had a shape of $(12, 422, 1)$, followed by three convolutional blocks, with 24, 48, and 48 filters respectively, *kernel* sizes of $(5, 5)$, $(2, 2)$, $(3, 3)$, convolutional layers' stride sizes of $(1, 1)$, and *max pooling* layers in the first two blocks, with $(2, 4)$, $(1, 3)$ *pooling* sizes, and stride sizes of the same dimensions respectively. After the last convolutional block, the data was flattened into a 1-D vector with $1,536$ elements, to be fed into 2 fully connected (FC) layers, with 64 and 6 neurons, respectively, totaling almost 125k free parameters. For regularization, a *dropout* of 0.5 was used before each FC layer. ReLU was used as the activation function for all layers, except for the output one, where *softmax* was used, to obtain the probability associated with each emotion.

3.3. 1-D CNN model

To assess the performance of 1-D CNN architectures, we reshaped the chromagram feature to match a model with an input length of 5064 samples. Following this, 3 convolutional blocks were added, with 16, 32 and 32 filters respectively, and kernel size of 4. The *max pooling* operation was implemented for all blocks with a *pooling* size of 3, and a stride of 1. The data outputted by the last convolutional block was then flattened, before being passed on to 3 FC layers with 1024, 128 and 6 neurons, respectively, totaling over 6M free parameters. Regularization and activation functions were maintained as in the 2-D CNN model.

3.4. Convolutional recurrent neural network (CRNN) model

Finally, we also experimented with the addition of a Bi-LSTM block after the convolutional layers of the model described in Section 3.3, aiming to learn with the long short-term temporal dependencies of audio signals. The convolutional blocks have the same parameters as described before, except for the number of filters, which are 16 in every block. The bidirectional wrapper takes an LSTM layer as argument, with 100 memory units, using ReLU as activation function, and a *dropout* of 0.5 as regularization. Ultimately, the Bi-LSTM layer output is flattened and passed on to 3 FC layers, with the same characteristics detailed in Section 3.3, totaling over 19M free parameters.

3.5. Data augmentation

In this work, we also explored suitable data augmentation (DA) techniques, i.e., creating slightly modified new data derived from original data. Since ANNs consider these new data as genuine, they can benefit from it, learning new parameters to achieve even better performances, without over-fitting. For this reason, we used the `Audiomentations` library, to add Gaussian noise to the original audio samples, and also shift the song's pitches, by 1 or 2 octaves. Models trained without DA were trained for 100 epochs, while models trained with DA were trained for 200 epochs.

4. Results and discussion

To train our models, the data-set was split into 612 samples for training, 200 for validation and 200 for test. In a previous work ([dos Santos et al. 2021]), it was already evinced that 2-D CNN models can surpass the overall test accuracy of MLP models. Therefore, our MLP model was trained only without data augmentation, first using the concatenation of 5 features described in Section 3.1, and then in a feature ablation manner, as detailed in Table 1. Next, our 2-D CNN model was trained with and without DA, to compare its performance under these different conditions. Since the use of DA improved the system's metrics, as evinced in Table 1, we continued using DA for the training of our 1-D CNN and CRNN architectures, always saving the epoch that achieved the best validation accuracy. Results show that the 1-D and 2-D CNN models achieved the best performances compared to all other models, while taking less time to train, compared with our CRNN model.

Moreover, we also resorted to the works of [Cunningham et al. 2018], [Yadav and Vishwakarma 2020] and [Atmaja and Akagi 2020] to compare our results with other works that used the same data-set, but not necessarily the same features and model architectures.

Tabela 1. Comparative results for different audio features and ANN models, using the song portion of the RAVDESS data-set.

Model	Features	Val. acc.	Val. loss	Test acc.	Test loss	Tr. time
MLP♥	Chromagram	0.78±0.07	1.5±0.01	0.77±0.03	1.54±0.08	18.24s
	MFCC	0.70±0.03	3.71±0.11	0.70±0.01	3.04±0.97	28.06s
	Spec. centroid	0.44±0.02	3.15±1.11	0.44±0.02	3.17±1.27	3.13s
	Spec. roll-off	0.45±0.01	2.91±1.01	0.45±0.02	2.91±0.95	3.03s
	ZCR	0.43±0.03	2.99±0.25	0.38±0.07	3.03±0.48	3.18s
	5 feats. concat.	0.65±0.06	2.5±0.76	0.65±0.09	2.66±1.74	5.14min
2-D CNN♥	Mel spec.	0.78±0.01	0.87±0.17	0.78±0.07	0.79±0.11	16.74s
	MFCC	0.70±0.01	0.81±0.04	0.63±0.02	0.86±0.01	7.74s
	Chromagram	0.84±0.01	0.49±0.02	0.80±0.01	0.62±0.02	8.42s
	Chromagram / DA	0.80±0.01	0.62±0.05	**0.84**±0.02	0.57±0.08	17.66s
1-D CNN♥	Chromagram / DA	0.87±0.02	0.42±0.07	0.83±0.01	**0.42**±0.05	44.82s
1-D CNN + BiLSTM♥	Chromagram / DA	0.81±0.03	0.81±0.45	0.75±0.02	0.89±0.44	11.21min
Logistic Regression♦	MATLAB Audio Analysis Library	-	-	0.48	-	-
RNN (LSTM)♦	Librosa HSF	-	-	0.82	-	-
1-D CNN + BiLSTM♣	MFCC / DA	-	-	0.73	-	-

Results associated with: ♥**ours**, ♦[Cunningham et al. 2018], ♠[Atmaja and Akagi 2020] and ♣[Yadav and Vishwakarma 2020]. All our models were trained using k-fold cross-validation, from which the mean and standard deviation values shown were computed.

According to [Priore and Stover 2014], harmony can be effectively used in song-writing to encode hidden meanings, e.g., to imprint emotiveness regardless of rhythm, lyrics etc., which is corroborated by our findings, in terms of accuracy. However, according to [Yadav and Vishwakarma 2020] and [Atmaja and Akagi 2020], we were expecting that adding a Bi-LSTM cell would improve the accuracy of our 1-D CNN model, since a same musical scale sang in different orders can imprint different moods. In spite of that, our CRNN model was significantly outperformed by our 1-D and 2-D CNN models, with DA. In terms of training time, our CRNN model was also outperformed by our 1-D and 2-D CNN models, and while our MLP models took even less time to train, they also produced lower accuracy in comparison. Looking into the learning curves of all 12 combinations of audio features and ANN models tested, as illustrated in Figure 2, it's observable that, regarding the MLP models (without DA), the ones with chromagram and MFCC as front-end start to diverge at around epoch 20, and start to over-fit at around epoch 60. On the other hand, the ones with spectral centroid, spectral roll-off and ZCR as front-end start to diverge early in their training, and over-fitting at around epoch 20, which is consistent with their poor performances. Using 5 features concatenated as front-end, the MLP model achieves its best accuracy, which can be justified by its late divergence. Regarding the 2-D CNN models, the ones with chromagram and Mel spectrogram as front-end start to diverge early in their training, yet stabilizing at high values of validation accuracy earlier as well. With DA, the 2-D CNN model stabilizes even earlier than others, which justifies its good performance. Finally, regarding the 1-D CNN models, it's noticeable that, without the Bi-LSTM layer, the model stabilizes at higher values of validation accuracy, without ever fully over-fitting, contrary to it's CRNN counterpart.

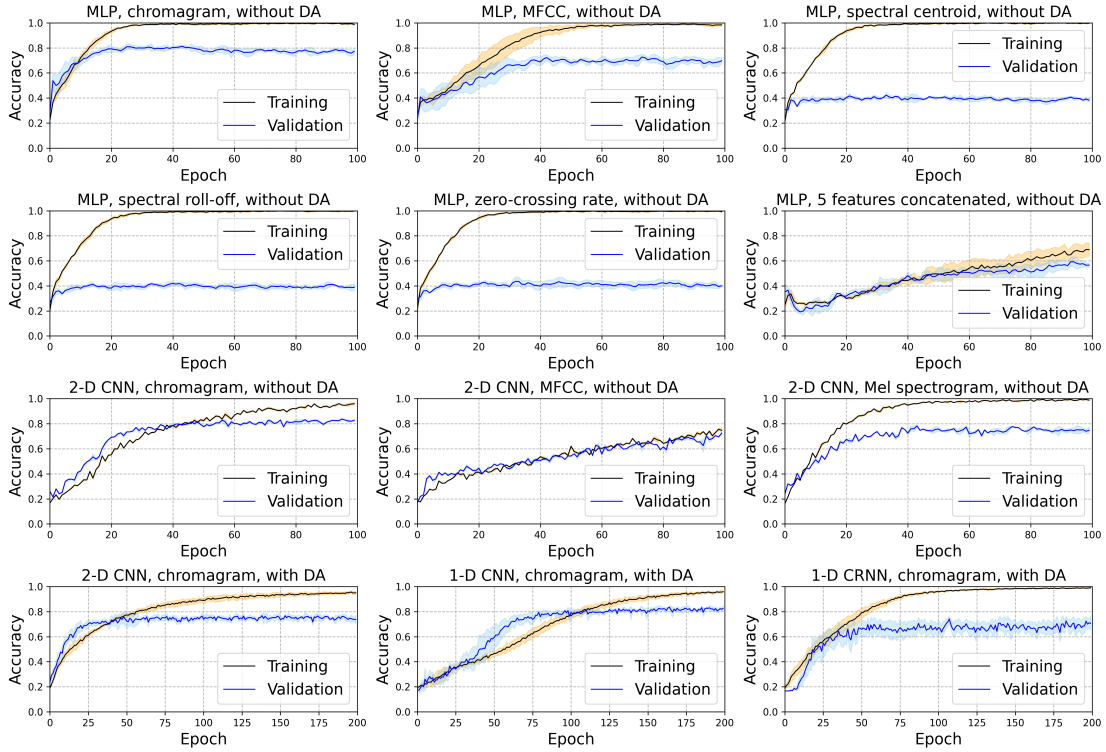

Figura 2. Learning curves for all 12 combinations of features and models tested. Shadows enveloping the curves represent standard deviations.

5. Conclusions

Although the most popular computational features for MER used in recent publications relate to tone color, the chromagram, which relates to harmony, was found to be best suited for song emotion recognition, in our experiments. Also, our 1-D and 2-D CNN models performed better than both our MLP and CRNN models, despite earlier works showing that RNN layers should improve the accuracy of 1-D CNN models. Moreover, our best result (0.84 ± 0.02 test accuracy) was obtained using data augmentation, evincing that ANNs do benefit from this technique. Ultimately, regarding song emotion recognition, our results are *state-of-the-art*, compared with recent publications, but it's important to mention that the data-set used in our experiments is far from being a general representation of the human population. Since it's samples lack accent, language, ethnical and gender diversities, disability inclusion etc., when put to test with samples different from its own *corpus*, it will hardly produce exciting results, such as in Table 1.

Referências

Atmaja, B. T. and Akagi, M. (2020). On the differences between song and speech emotion recognition: Effect of feature sets, feature types, and classifiers. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 968–972.

Casper, L. (2020). Creating a speech and music emotion recognition system for mixed source audio. Master's thesis, Universiteit Utrecht.

Cunningham, S., Ridley, H., Weinel, J., and Picking, R. (2020). Supervised machine learning for audio emotion recognition. *Personal and Ubiquitous Computing*.

Cunningham, S., Weinel, J., and Picking, R. (2018). High-level analysis of audio features for identifying emotional valence in human singing. In *In: Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, pages 1–4.

dos Santos, A. N., Jácome, K. G. R., and Masiero, B. S. (2021). Song emotion recognition: A study of the state of the art. *Anais do XVIII Simpósio Brasileiro de Computação Musical*, pages 209–212.

Du, P., Li, X., and Gao, Y. (2020). Dynamic music emotion recognition based on cnn-bilstm. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1372–1376.

Flamia Azevedo, B. and Bressan, G. (2018). A comparison of classifiers for musical genres classification and music emotion recognition. *Advances in Mathematical Sciences and Applications*, pages 241–262.

Gao, Z., Qiu, L., Qi, P., and Sun, Y. (2020). A novel music emotion recognition model for scratch-generated music. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1794–1799.

Kim, W. (2020). Musemo: Express musical emotion based on neural network. Master's thesis, Ulsan National Institute of Science and Technology.

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., and Turnbull, D. (2010). Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952.

Livingstone, S. R. and Russo, F. A. (2018). 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).

Ospitia Medina, Y., Beltrán Blázquez, J. R., and Baldassarri, S. (2020). Emotional classification of music using neural networks with the mediaeval dataset. *Personal and Ubiquitous Computing*.

Panda, R., Malheiro, R. M., and Paiva, R. P. (2020). Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, pages 1–1.

Pandrea, A. G., Gómez-Cañón, J. S., and Herrera, P. (2020). Cross-Dataset Music Emotion Recognition: an End-to-End Approach.

Priore, I. and Stover, C. (2014). The subversive songs of bossa nova: Tom jobim in the era of censorship. *Analytical Approaches to World Music*, 3(2):1–32.

Rajesh, S. and Nalini, N. J. (2020). Musical instrument emotion recognition using deep recurrent neural network. *Procedia Computer Science*, 167:16–25. International Conference on Computational Intelligence and Data Science.

Russo, M., Kraljević, L., Stella, M., and Sikora, M. (2020). Cochleogram-based approach for detecting perceived emotions in music. *Information Processing & Management*, 57(5):102270.

Yadav, A. and Vishwakarma, D. K. (2020). A multilingual framework of cnn and bi-lstm for emotion classification. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6.