Song Emotion Recognition: A Study of the State of the Art

Arthur Nicholas dos Santos[1]*, Karen Gissell Rosero Jácome[1]†, Bruno Sanches Masiero[1].

[1]Laboratório de Acústica das Comunicações (LAC) – Faculdade de Engenharia Elétrica
e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP)
Av. Albert Einstein, 400, Bloco A – 13083-852 Campinas, SP

a264372@dac.unicamp.br, k264373@dac.unicamp.br, masiero@unicamp.br

Abstract. *Music is art, and art is a form of expression. Often, when a song is composed or performed, there may be an intent by the singer/songwriter of expressing some feeling or emotion through it, and, by the time the music gets in touch with an audience, a spectrum of emotional reactions can be provoked. For humans, matching the intended emotion in a musical composition or performance with the subjective perceptiveness of different listeners can be quite challenging, in account that this process is highly intertwined with people's life experiences and cognitive capacities. Fortunately, the machine learning approach for this problem is simpler. Usually, it takes a data-set, from which features are extracted to present this data to a model, that will train to predict the highest probability of an input matching a target. In this paper, we studied the most common features and models used in recent publications to tackle music emotion recognition, revealing which ones are best suited for songs (particularly acapella).*

1. Introduction

Music Emotion Recognition (MER) is a sub-field of Music Information Retrieval (MIR) that deals with classification of music according to affection [1]. The importance of MER can be justified by the dependency that search and recommendation engines have on metadata. In simple terms, metadata is data about data. For example, when a person takes a picture of a cat using a smartphone, the picture is the data itself. However, a series of other information about that data is also recorded, e. g., the time and date when the picture was taken, or the geographical coordinates where it was captured. These are all secondary information about a primary one, that can be used to tag it, in order to retrieve it in the future, as well as finding other information that is similar to it.

Song metadata like genre, composer, artist, album, year of release etc. are commonly used by streaming services to help users find what they like, and even recommend songs they might like based on their listening history. However, the mood of a song can also be considered an interesting metadata that could be used to relate a certain song to similar content.

This paper studies various recent articles published on MER, including papers on song and instrumental music emotion recognition, and is organized as follows: Section 2 details our findings on what are the most commonly used features for representing music samples in order to present this type of data to a machine learning (ML) model, and covers our findings on the most commonly used model architectures. Section 3 describes our experiment, which consists in creating a couple of ML models based on the information retrieved on the previous sections, as well as the data-set used to test our models. Section 4 shows our results in terms of accuracy, comparing it to previous works that used the same data-set. Finally, Section 5 draws some conclusions on what are the best combination of features and models for song emotion recognition that we found.

*Grant #2019/22795-1, São Paulo Research Foundation FAPESP.
†Grant #2019/22945-3, São Paulo Research Foundation FAPESP.

2. Features and Models

According to Panda, Malheiro and Paiva, 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 features may be identified [2].

By reviewing 10 articles on MER published in 2020 alone we have found 47 different low-level computational features being used separately or concatenated, to represent different aspects of the aforementioned high-level features [3–12] . 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:

Spectral roll-off: relates to tone color and indicates the frequency below which approximately 85% of the magnitude spectrum distribution is concentrated [2]. Was used in [3], [5], [8–10], and [12].

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 [2]. Was used in [3], [5], [8–10], and [12].

Spectral centroid: also relates to tone color and represents the mean of the magnitude spectrum of the short-time Fourier Transform (STFT) [2]. Was used in [3–5], [8, 9], and [12].

Mel spectrogram: also relates to tone color 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 [2]. Was used in [4–8], and [11].

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 Fourier Transform (DCT). Typically, only the first 8 to 13 MFCCs are used for speech representation [2]. Was used in [3–5], [9, 10], and [12].

Chromagram: relates to harmony and indicates energy distribution along pitch classes in a 12-dimensional vector (12 semitones, from A to G#) [2]. Was used in [4, 5], [8–10], and [12].

As for the other 41 features found (which were used on only 23.4% of the publication reviwed), 13 of them are related to rhythm, 10 are related to tone color, 6 are related to harmony, 5 are related to melody and dynamics (each), and only 1 is related to texture, form and vocals (each).

Depending on the architecture of the ML model used as a classifier for MER, several of the aforementioned front-end features can be used together, to better represent the training data. However, not all models allow that. According to de Azevedo and Bressan, what dictates which and how many features can be used as front-end for an ML model is the architecture of the model itself [3]. In our study, 12 different architectures were found to be used, separately or combined, and 3 of them were found to be used on 17% of the publications reviewed (each):

Support Vector Machine (SVM): is a binary classifier that divides the training data into groups by using hyper-planes. An SVM finds an optimal hyper-plane by using the dot product functions in feature space using kernel functions. The solution of the optimal hyper-plane can be written as a combination of a few input points that are called support vectors [3]. Was used in [3], [6], [8, 9] and [12].

Multi-Layer Perceptron (MLP): is an artificial neural network (ANN) that models the relationship between a set of training data and 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 relation between input and output data is well understood, yet the process that relates both is extremely complex [3]. Was used in [3], [5], [8, 9] and [12].

Convolutional Neural Network (CNN): is a type of ANN based on convolutional operations that can extract high-level features from 2-dimensional low-level ones, such as Mel-spectrogram, MFCCs and chromagram. It deeply extracts underlying features contained in each time frame, while retaining time-series features in the same direction. After a CNN block, a fully connected MLP block is often used to predict outputs in classification problems [4]. Was used in [4], [6, 7], and [11, 12].

As for the other 9 architectures, recurrent neural networks (RNN) with long-term short-memory (LSTM) blocks and random forest were found to be used on 10% of the publications reviewed (each), k-nearest neighbors 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 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 features and models described in Section 2, a portion of the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) data-set was chosen. This portion comprises 1,012 audio-only files of song recordings, performed by 23 actors singing 2 lexically matched statements in a neutral North American accent. Song emotions include neutral, calm, happy, sad, angry, and fearful expressions [13].

Since an MLP model can have an input layer with as many neurons as necessary, all input features can be concatenated and flattened into a 1-dimensional input vector. Therefore, we used the principal component analysis (PCA) technique to reduce the dimensionality of each of the 6 features presented in Section 2, aiming to visualize the minimum number of variables that keeps the maximum amount of information about how the feature data is distributed, as illustrated in Figure 2. Mel spectrogram was discarded because it showed the worst clustering and the highest dimensionality compared to the other features, moreover we included MFCCs that are already computed based on it. Hence, we chose to train an MLP model using the concatenation of the 5 selected features.

Table 1 summarizes the architecture of our MLP model, which has an input layer with 11,394 neurons, followed by 2 hidden layers with 1,024 and 128 neurons, respectively, totaling 141,635,084 trainable parameters. ReLU was used as an activation function for all layers, except for the output layer, where Softmax was implemented. For regularization, Dropout was used, which randomly ignores a percentage of neurons during training.

Table 1: Summary of our MLP model

Layer type	Output shape	Param. #
Dense	(None, 11,394)	129,834,630
Dropout	(None, 11,394)	0
Dense	(None, 1,024)	11,668,480
Dropout	(None, 1,024)	0
Dense	(None, 128)	131,200
Dropout	(None, 128)	0
Dense	(None, 6)	774

For our CNN model, we selected the 2-dimensional feature that showed the clearest clustering in the PCA visualization (Figure 2), which was the chromagram. Table 2 summarizes the architecture of our CNN model, which has an input shape of $(12, 422, 1)$. To obtain the output for the second convolutional layer, 24 kernels with a shape of $(5, 5)$ were used, with a stride of $(1, 1)$. Another typical CNN operation, which is Pooling, was used to down-sample data and compress information, with a shape of $(2, 4)$ and a stride of $(2, 4)$. The third convolutional layer used 48 kernels with a shape of $(2, 2)$, followed by a Pooling operation, with a shape of $(1, 3)$ and

a stride of $(1, 3)$. After the last convolutional layer, the data was flattened into a 1-dimensional vector with 1,536 elements, to be fed into 2 fully connected layers, with 64 and 6 neurons, respectively, totaling 124,822 trainable parameters. Dropout was used for regularization and ReLU as activation function for all layers, except for the output layer, where Softmax was used.

Table 2: Summary of our CNN model

Layer type	Output shape	Param. #
Conv2D	(None, 8, 418, 24)	624
MaxPooling2D	(None, 4, 104, 24)	0
Activation	(None, 4, 104, 24)	0
Conv2D	(None, 3, 103, 48)	4,656
MaxPooling2D	(None, 3, 34, 48)	0
Activation	(None, 3, 34, 48)	0
Conv2D	(None, 1, 32, 48)	20,784
Activation	(None, 1, 32, 48)	0
Flatten	(None, 1,536)	0
Dropout	(None, 1,536)	0
Dense	(None, 64)	98,368
Activation	(None, 64)	0
Dropout	(None, 64)	0
Dense	(None, 6)	390
Activation	(None, 6)	0

Since we were dealing with a multi-classification problem, and an SVM is a binary classifier, we opted to focus on the two architectures already discussed (MLP and CNN) and do not implement an SVM model.

4. Results and Discussion

To train both our models, the data-set was split into 622 samples for training, 200 for validation and 200 for test. Both our models were trained for 100 epochs, however, Figure 1 illustrates that the MLP learning curves begin to diverge at around epochs 50-55, while the CNN learning curves begin to diverge at around epochs 25-30. This shows that the CNN model achieve its best results in less training epochs, compared to the MLP model.

On the left hand side of Figure 2, PCA plots are illustrated for the concatenated features (top) and the chromagram alone (bottom), both before training, exposing mixed clusters. On the right hand side, it is shown that the samples are nicely spaced and grouped together after the classification performed by our MLP (top) and CNN (bottom) models. This evinces that the use of ML architectures together with a good feature selection is a favorable technique for MER.

The overall test accuracy of the MLP model achieved only 67.7%, while the CNN achieved 80.6%. Moreover, Figure 4 shows that the MLP test accuracy per class ranges from 6.8% (happy) to 16.7% (angry), while the CNN test accuracy per class has a worst-case scenario of 11.2% (happy) and best-case scenario of 16.7% (fearful).

We resorted to the works of [14–16] to compare our results with papers that used the same data-set, but not

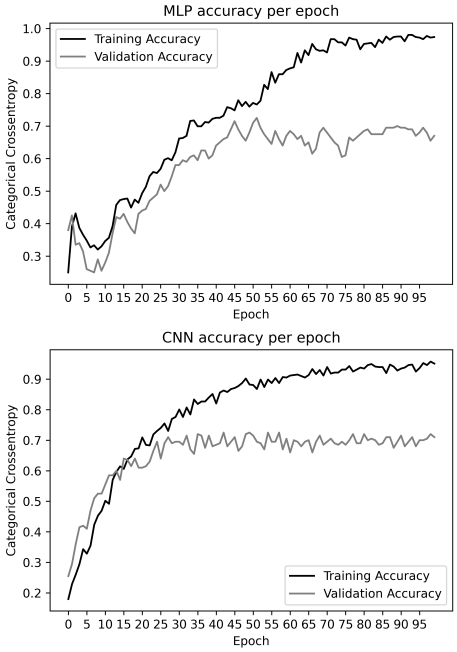

Figure 1: MLP and CNN learning curves

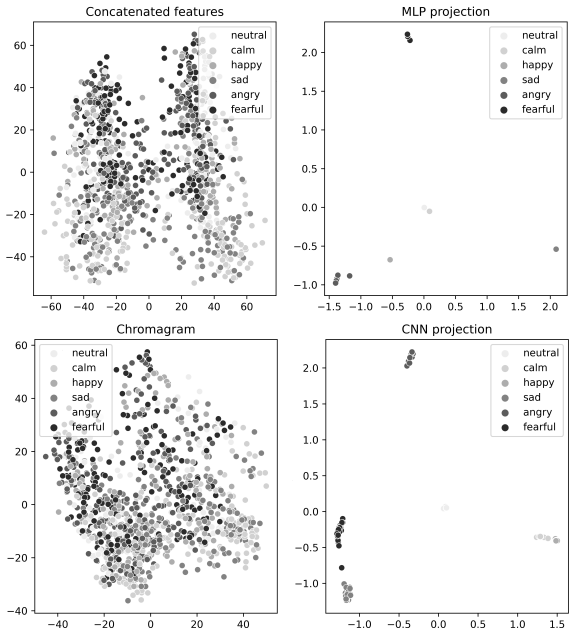

Figure 2: PCA visualization for MLP and CNN

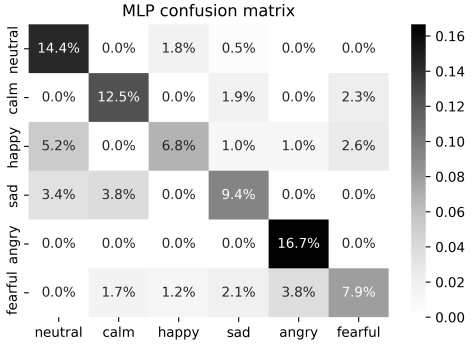

Figure 3: MLP confusion matrix using 5 features

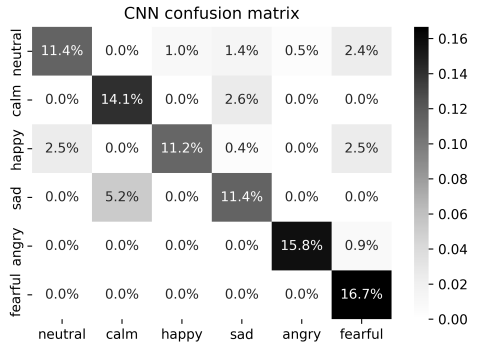

Figure 4: CNN confusion matrix using 1 feature

necessarily the same features and model architectures. Table 3 summarizes each work's authors, features and architectures used, as well as the overall test accuracy obtained.

Table 3: Accuracy comparison with other works

Authors	Features	Models	Acc.
[14]	MATLAB Audio Analysis Library	Logistic Regression	48%
Ours	**Concatenated features**	**MLP**	**67.7%**
Ours	**Chromagram**	**2D CNN**	**80.6%**
[16]	LibROSA HSF	RNN (LSTM)	82%
[15]	MFCC	1D CNN +BiLSTM	94%

We also performed a forensic analysis on our MLP model, to see which features yield the best performance. Using only the chromagram, the overall accuracy achieved was equal to 66.7%, while using only MFCC it was 45.9%. Spectral roll-off, ZCR and spectral centroid yielded the worst performances, with 29.6%, $28,5\%$ and 24.7% overall test accuracy, respectively.

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 CNN model performed better than our MLP model, both in terms of overall test accuracy and accuracy per-class, because our MLP model favored 3 emotions (neutral, calm and angry), while causing more confusions for the predictions of the other 3 emotions (happy, sad and fearful). Finally, it was evinced by comparison with the works of [15, 16] that CNN performance can be further improved with the use of RNN blocks, that can learn order dependence in sequence prediction problems, which may be a desired behavior in song emotion recognition.

References

[1] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952, 2010.

[2] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.

[3] Beatriz Flamia Azevedo and Glaucia Bressan. A comparison of classifiers for musical genres classification and music emotion recognition. pages 241–262, January 2018.

[4] Zijing Gao, Lichen Qiu, Peng Qi, and Yan Sun. A novel music emotion recognition model for scratch-generated music. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1794–1799, June 2020.

[5] Laugs Casper. Creating a speech and music emotion recognition system for mixed source audio. Master's thesis, August 2020.

[6] Mladen Russo, Luka Kraljević, Maja Stella, and Marjan Sikora. Cochleogram-based approach for detecting perceived emotions in music. *Information Processing & Management*, 57(5):102270, September 2020.

[7] Wooyeon Kim. Musemo: Express musical emotion based on neural network. Master's thesis, February 2020.

[8] Ana Gabriela Pandrea, Juan Sebastián Gómez-Cañón, and Perfecto Herrera. Cross-Dataset Music Emotion Recognition: an End-to-End Approach, 2020.

[9] Pengfei Du, Xiaoyong Li, and Yali Gao. Dynamic music emotion recognition based on cnn-bilstm. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1372–1376, 2020.

[10] Yesid Ospitia Medina, José Ramón Beltrán Blázquez, and Sandra Baldassarri. Emotional classification of music using neural networks with the mediaeval dataset. *Personal and Ubiquitous Computing*, April 2020.

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

[12] Stuart Cunningham, Harrison Ridley, Jonathan Weinel, and Richard Picking. Supervised machine learning for audio emotion recognition. *Personal and Ubiquitous Computing*, April 2020.

[13] Steven R. Livingstone and Frank A. Russo. 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), 2018.

[14] Stuart Cunningham, Jonathan Weinel, and Richard Picking. 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, September 2018.

[15] Ashima Yadav and Dinesh Kumar Vishwakarma. 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, 2020.

[16] Bagus Tris Atmaja and Masato Akagi. 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, 2020.