Predicting Perceived Music Emotions with Respect to Instrument Combinations

Viet Dung Nguyen [*1], **Quan H. Nguyen** [*2], **Richard G. Freedman** [3]

[1] Rochester Institute of Technology
[2] Gettysburg College
[3] SIFT
vn1747@rit.edu, nguyqu03@gettysburg.edu, rfreedman@sift.net

Abstract

Music Emotion Recognition has attracted a lot of academic research work in recent years because it has a wide range of applications, including song recommendation and music visualization. As music is a way for humans to express emotion, there is a need for a machine to automatically infer the perceived emotion of pieces of music. In this paper, we compare the accuracy difference between music emotion recognition models given music pieces as a whole versus music pieces separated by instruments. To compare the models' emotion predictions, which are distributions over valence and arousal values, we provide a metric that compares two distribution curves. Using this metric, we provide empirical evidence that training Random Forest and Convolution Recurrent Neural Network with mixed instrumental music data conveys a better understanding of emotion than training the same models with music that are separated into each instrumental source.

1 Introduction

Emotion is an important aspect of humans that can have paramount effects on humans' motivation. Most of our daily actions and behaviors are inspired by mood and emotion. Humans also express emotions consciously or unconsciously across time. There are many ways we can express emotion: through literature, poem, film, conversation, or facial expression. Music is also a means of transforming the intensity and variety of emotions to others. In music, we can hear the happiness in a ballad love song, we can hear the nostalgia in a slow monotone song, and we can also hear the joy in a steady, fast-tempo song. Music itself expresses emotion, and some of the emotions are believed to be perceived more than others (Juslin 2013).

We are living in a high-technology world where we have computing power from modern processors that automate many human tasks. In the context of music emotional expression, one can utilize the enormous computational power to classify and group songs with the same emotional scheme together, which can benefit people listening to music. Another application of machines in automating music emotion prediction is to recommend appropriate songs that match the current listener's mood. Because we see that machines can be utilized in the prediction of music emotions, and such work can be potentially applied in many fields, we decided to conduct research on using machines (and machine learning algorithms) to learn and predict the emotion in music.

However, labeling emotion is a difficult problem because some of our emotion is deep inside the cognitive unconsciousness, unarticulated, and linguistically inaccessible (Izard 2008). In fact, there are a number of instances where there is a high variance of emotional frequency. Particularly, happiness, sadness, anger, fear, love, and tenderness are among the top-rated emotions when people listened to a collection of designated songs (Juslin 2013). Because of such reasons, we are motivated to study the accuracy of emotion annotation and find metrics that can compare different emotion predictions to each other.

In this paper, we focus on how machines perceive music emotions. Diving deeper, we were amazed by the articles that claim our perception of emotion in music is affected by the timbre (instrument identity). We found Hailstone et al. (2009)'s *Experiment 1* interesting. Depending on which instrument is played (piano - percussion timbre, violin - string timbre, trumpet - brass timbre, and electronic synthesizer - "electronic" timbre), people can feel different emotions even for the same notes and melody, but the intended emotion was created through the note and melody choices when writing the piece of music. Therefore, we investigate whether separating each instrument's part of a song as multiple individual waveforms could help improve the performance of machines understanding emotion in music in ways that people do.

We hypothesize that training a model with separate instruments will better predict the song's emotion that humans perceive than training the model with all instruments together. In order to test this hypothesis with both Convolution Recurrent Neural Network and Random Forest, we train a model of each type for each dataset and compare the statistics of the results with the metric we produce. Our key contributions are procedures for training using songs with separated instruments and an evaluation method that considers the variance of people's emotions. We discuss future work based on our results, which rejected this hypothesis.

2 Background

People can perceive any emotion from a piece of music, and we cannot accuse that listener of being wrong. Besides our

[*] These authors contributed equally.

personal perception of music, mood can be affected by various factors such as musical style, response format, or procedure. Hence, the *precision of emotion conveyed by music is limited*. To label emotion perceived from music, researchers broadly sample emotions that listeners perceive (Juslin and Laukka 2004). As a result, there are not a lot of refined datasets constructed, which presents a challenge for the Music Information Retrieval (MIR) field. Especially for Music Emotion Recognition (MER) tasks, to assess the emotion of the song, one has to collect the songs as input (most of them are not possible because of copyright restrictions). According to Aljanaki, Yang, and Soleymani (2017), emotion is subjective to humans and languages; therefore, they are difficult to determine.

Different datasets have their own labeling scheme. There are a lot of emotion labeling schemes such as the emotion adjective wording scheme from Lin, Yang, and Chen (2011) or the two-dimensional regression scheme from the datasets such as in Aljanaki, Yang, and Soleymani (2017) and Zhang et al. (2018), which utilize the two orthogonal psychology states that are proposed by Russell (1980). These two orthogonal states are valence (i.e., the positiveness of the song) and arousal (i.e., the energy of the song). Many datasets are built for the music emotion recognition task using this labeling scheme. For instance, the DEAM dataset (Aljanaki, Yang, and Soleymani 2017) contains 1,802 western pop excerpts and corresponding valence-arousal annotations. Another dataset that uses this scheme is DEAP (Koelstra et al. 2012), which also uses valence-arousal rating as music emotion evaluation.

Regarding audio features, Purwins et al. (2019) mentioned that Mel Frequency Cepstral Coefficients (MFCCs) used to be utilized broadly as input. However, since MFCCs cause information loss and lack spatial relations, they are considered unnecessary as deep learning models become increasingly popular. In contrast, spectrograms (log-mel spectrum, or constant-Q spectrum) are an image-based feature of audio that represents the correlation between time and audio frequency; it has been used widely in predicting music emotion with Convolution Neural Network (CNN) (Liu et al. 2017; Yang et al. 2020; Chowdhury et al. 2019; Dong et al. 2019). There are other works that extract the low-level acoustics feature of the song such as pitch, loudness, and cepstrum (Eyben, Wöllmer, and Schuller 2010). Those features are grouped and concatenated into a long feature vector.

Researchers have tried various methods for MER tasks. According to a survey by Han et al. (2022), there are two branches in MER. One branch, static MER, recognizes the overall emotion of the song; static MER is usually a multi-class classification problem. The other branch, dynamic MER, classifies emotions based on the instantaneous moments throughout the song. Badshah et al. (2017) used a CNN to estimate the spectrogram features that represent the sounds related to speech in order to predict the text corresponding to that sound. Vempala and Russo (2018) also proposed a shallow Neural Network (Meta Cognitive Network) whose input features used Principal Components Analysis to reduce their dimension.

3 Methods

In this section, we first describe the dataset and techniques we used to extract features from the raw music waveform. We then discuss the Convolution Recurrent Neural Network and Random Forest models in detail.

Dataset

Past MER research concluded that there are several factors that differentiate the emotion described by the song from the emotion felt by humans (Gabrielsson 2002; Song et al. 2016). Therefore, we must consider which type of emotion was assessed in a dataset and determine whether it aligns with the type of emotion we intend to study. In order to describe the difference between training machine learning models on music with all instruments at once or separate instruments, we investigated the datasets containing *human-perceived emotion* annotations because we want to investigate whether there are any differences in *describing the song's emotion* as a whole versus as a combination of separated instrumental sources.

In the scheme of this research, we choose to conduct our experiment on the PMEmo dataset (Zhang et al. 2018). The dataset consists of 794 45-second-long pieces of music in the raw waveform format. The emotions are annotated according to Russell's emotional scheme (Russell 1980). In particular, participants were given a platform where they could grade two emotional dimensions–valence and arousal–on a 9-point Likert scale that was then normalized from 0 to 1. There were a total of 457 subjects in the PMEmo dataset's annotation process, and their annotations per song are aggregated in the form of a *mean and standard deviation for each emotional dimension*. The individual subject's responses are not included in the dataset.

We used a pre-trained model from Wave-U-Net (Stoller, Ewert, and Dixon 2018) to decompose each song in the PMEmo dataset into a group of four separate raw instrumental waveforms (bass, drums, vocal, and others). Originally, the Wave-U-Net (Stoller, Ewert, and Dixon 2018) model was trained on MUSDB18 (Rafii et al. 2017) for instrumental accompaniments and CCMixter (Liutkus, Fitzgerald, and Rafii 2015) for music with lyrics training. We also expect that there is a domain gap between training datasets (i.e., datasets trained with Wave-U-Net) and the evaluation dataset (i.e., PMEmo dataset). Particularly, there can be a covariate shift between MUSDB18 and PMEmo dataset (i.e., the MUSDB18 dataset's collection music cannot generalize well enough for all waveform inputs due to representation bias). Additionally, the MUSDB18 dataset has an imbalance among its label instruments. That is, the 'Vocal', 'Bass', and 'Drum' labels are separated as standalone instruments, but the 'Other Instruments' label contains everything else without separating them. We acknowledge the dataset shift and imbalance concerns when we apply this particular Wave-U-Net model to the music samples in the PMEmo dataset. In this experiment context, we denote "mixed" for the original PMEmo dataset waveform with original sound sources (mixed instrument), and "sep" for the dataset in which each song's waveform in the PMEmo dataset is separated into

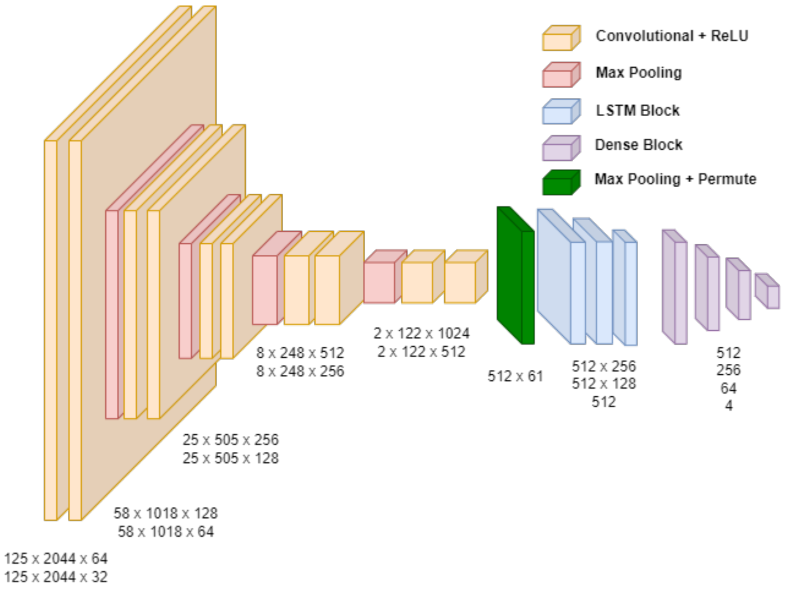

Figure 1: Convolution Neural Network with Long Short-term Memory Heads

four instruments (vocal, bass, drum, and others) using the Wave-U-Net pre-trained model.

Feature Extraction

We must extract appropriate features from the waveform data to estimate a song's emotion. The choice of features directly affects the model's prediction capabilities.

Several related research works used spectrograms as their input feature to the convolutional training system (Liu et al. 2017; Yang et al. 2020; Chowdhury et al. 2019; Dong et al. 2019). In the scheme of training the Convolution Recurrent Neural Network model, we chose to use the spectrogram feature that was computed by Short-time Fourier Transform with input from the raw waveform of the music. Originally, the raw music waveform is 1-dimensional data that describes how many decibels a particular sample at a time step has (e.g., how loud the sample is at one point in time). With a Short-time Fourier Transformation of that data to Spectrogram data, the feature becomes 2-dimensional as with each point in time, there is a vector of features that describe the energy at each audible sound frequency.

For the Random Forest training system, we were inspired by Schuller et al. (2016), who proposed a set of features for automatic paralinguistic phenomena detection. It was used in the PMEmo dataset (Zhang et al. 2018). We used OpenSMILE (Eyben, Wöllmer, and Schuller 2010) to extract features with the ComParE_2016 feature set, which extracts features with a 6373-dimension scale at OpenSMILE's Functionals level. We ran OpenSMILE on both mixed (original) and separated (vocal, bass, drum, and "other" instruments) waveform data.

Convolution Recurrent Neural Network

One of our chosen extracted features is the spectrogram representation of the raw waveform existing in the form of two-dimensional data, and the spectrogram has its image patterns varied across different frequencies in the time axis according to the different timbres of the sound (Badshah et al. 2017). Therefore, we believe that it makes sense for our model to reason from such features using an image feature extraction technique. Furthermore, past research in music emotion prediction also utilizes this image-like nature and uses convolution to extract the time-frequency features of the music (Malik et al. 2017; Hizlisoy, Yildirim, and Tufekci 2021). Thus, we chose to use multiple blocks of convolution layers to extract these features into a latent vector.

When using convolutions to interpolate and downsize the spectrogram features, the nature of the spectrogram's time axis does not change. When reducing the frequency axis of the feature to 1, we had a vector with the downsized time axis and a channel axis (i.e., represents the frequency features through neurons). At this stage, we had a matrix in which there is a latent vector of hidden features that represents the latent data at each timestamp. We used Long Short-term Memory (Hochreiter and Schmidhuber 1997) for the model to reason over the temporal data given the feature vectors at each point in time. The reason we chose Long Short-term Memory over Recurrent Neural Network is that it has multiple gates that add or remove weights from the memory cell, which helps the network retain both long-term and short-term memory in a long sequence such as audio waveform data. Additionally, Hizlisoy, Yildirim, and Tufekci (2021) also made use of the Long Short-term Memory in their music emotion prediction network.

After reasoning with the temporal knowledge to output a latent vector, we use a combination of linear (i.e, Dense Layer) and non-linear (i.e, Rectified Linear Unit) transformations to compute the result vector. This vector has four values corresponding to the input music's emotion prediction: mean valence, mean arousal, standard deviation of valence, and standard deviation of arousal.

Lastly, because we are fitting a regression model where we have to estimate these statistical parameters describing the song's emotion, we use the last layer as a normal linear transformation layer to compute loss by the Mean Square Error function. The derivative of the loss with respect to the weights of the model is then computed, and the model is fitted with an optimization function as in the backpropagation process (Rumelhart, Hinton, and Williams 1986). The detailed architecture of the network is described in Figure 1 and the Experiment section (Section 4).

Random Forest

It is necessary to determine and narrow down correlated features as a preprocessing step before performing a regression model. However, it is difficult to perform feature selection on the whole feature set because it has a large number of features, more than 6000 from the mixed dataset and more than 25000 from the separated dataset. Therefore, Random Forest is our choice since Cutler, Cutler, and Stevens (2011) stated that the Random Forest algorithm is robust against high-dimensional problems and less susceptible to multicollinearity. Furthermore, Random Forest is made up of multiple Decision Trees, which helps solve bias or overfitting problems (Breiman 2001).

Validation Method

To validate the models' performance, we rejected using Hold-Out Validation since Refaeilzadeh, Tang, and Liu suggested that it creates data in the training set that is not enough to train models, which causes skewed results on the testing data (Refaeilzadeh, Tang, and Liu 2009). Therefore, we used Monte Carlo Cross-Validation to avoid skewing our results (Xu and Liang 2001).

KL Divergence between Distributions

We propose a method for comparing the model accuracy when predicting valence and arousal distribution parameters. The PMEmo dataset only supplies the mean and standard deviation of all participants' emotion ratings for each particular song without describing the distributions' shapes. Without this information, we choose to *assume the distribution parameters define a multivariate normal distribution*. That is, we assume that the PMEmo dataset's emotion values per song across participants are normally distributed. The PMEmo dataset further provides valence and arousal means and standard deviations independently of each other (Zhang et al. 2018). Without a way to determine the covariance relationship between the valence and arousal, we also *assume that there is no covariance between the valence distribution and the arousal distribution per song*.

Combining our assumptions, the annotated emotion (i.e., valence and arousal) for each song would most likely form an ellipse-shaped region around a point (the means) whose axes represent the variance around it (the standard deviations). That is, we assume each song's emotion is a multivariate normal distribution. Therefore, KL Divergence, a measure of how two distributions are different from each other (Kullback and Leibler 1951), is suitable to measure how our models' predictions are different from the ground truth. We first obtained all the prediction results from evaluating each model with the same test set, and then we computed the KL Divergence that the predicted emotion distribution Q has from the ground truth emotion distribution P. As multivariate normal distributions, we can compute the KL Divergence as (Soch et al. 2020):

$$\begin{aligned} P &= \mathcal{N}(\mu_1, \Sigma_1) \\ Q &= \mathcal{N}(\mu_2, \Sigma_2) \\ KL[P||Q] &= \frac{1}{2}\Bigg[(\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1) \\ &\quad +\text{tr}(\Sigma_2^{-1}\Sigma_1) - \ln\frac{|\Sigma_1|}{|\Sigma_2|} - n\Bigg]. \end{aligned} \tag{1}$$

Note that in Equation 1, μ_1 and μ_2 represent the mean ground truth emotion and mean predicted emotion, respectively. μ_1 and μ_2 are vectors of length two with statistics: mean valence and mean arousal. n is the number of distribution dimensions, which is 2 in our case for valence and arousal. For each song, Σ_1 is the covariance matrix of the ground truth valence distribution and the ground truth arousal distribution and Σ_2 is the covariance matrix of the predicted valence distribution and the predicted arousal distribution. Within the variance notation σ^2, we use v for valence, a for arousal, gt for ground truth, and pr for prediction:

$$\Sigma_1 = \begin{pmatrix} \sigma^2_{gt,v} & 0 \\ 0 & \sigma^2_{gt,a} \end{pmatrix}; \Sigma_2 = \begin{pmatrix} \sigma^2_{pr,v} & 0 \\ 0 & \sigma^2_{pr,a} \end{pmatrix} \tag{2}$$

After computing the KL Divergence for all the test set's songs' emotion distributions, we can compare the performance of the models using "mixed" and "sep" audio waveform representations for each corresponding song. When a single KL Divergence for one model (e.g., model trained on "sep" dataset) is smaller than the other model's corresponding KL Divergence, it means that that model's emotion distribution prediction is closer to the ground truth emotion distribution for that song. Therefore, to see which model (either model trained on "mixed" or "sep" dataset) performs better (i.e., has a lower distribution distance to the ground truth overall), we can compute the mean E and median M statistics for all the KL Divergences that were computed in the test dataset for both "mixed" and "sep" dataset. For example, with m as the number of songs in a particular test dataset, and $KL[P||Q]_i$ as the particular KL Divergence at the song i, we have the following expected value formula:

$$E = \frac{1}{m}\sum_{i=1}^{m} KL[P||Q]_i \tag{3}$$

We also want to determine whether the "mixed" and "sep" models' KL Divergence distributions over all songs are significantly different from each other before concluding that

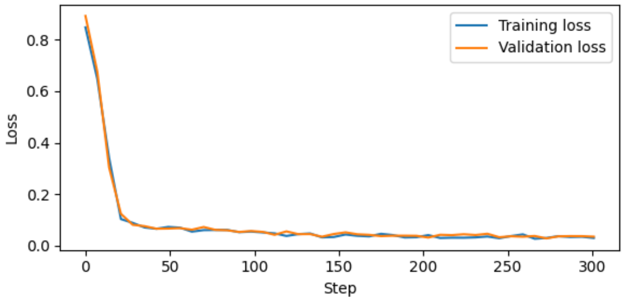

Figure 2: Convolution Recurrent Neural Network's mean training and validation loss among 10-iteration cross-validation for the model trained with "mixed" dataset.

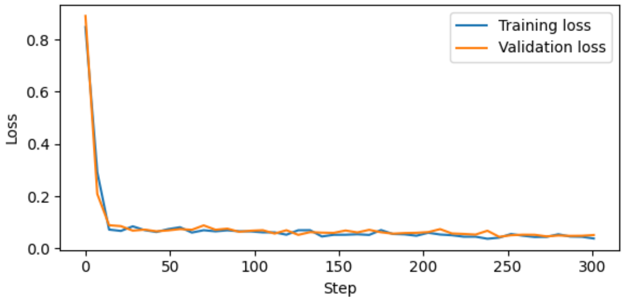

Figure 3: Convolution Recurrent Neural Network's mean training and validation loss among 10-iteration cross-validation for the model trained with "sep" dataset.

one model performs MER better. For each model, we aggregated each song's KL Divergences and plotted histograms of the KL Divergences to visualize their distribution across the songs in the test set. Since we saw that the KL Divergence distributions are not normal distributions (Figure 4), we use the Wilcoxon Signed Rank Test rather than the two-sample t-test. This test computes the probability that the two KL Divergence distributions for each model and cross-validation iteration are similar. A probability of 0 implies that the two distributions' means are totally different, and a probability of 1 implies that the two distributions' means are totally the same.

4 Experiment

Before the experiment, we needed to preprocess the dataset. We converted every waveform to 44.1 Khz in the ".wav" format. We then used the Wave-U-Net pre-trained model that was previously trained on the MUSDB18 dataset to segment the waveforms in the PMEmo dataset.

For the Monte Carlo Cross-Validation, we shuffled the songs in the dataset per iteration. After shuffling, we split the dataset into a training set and testing set with the ratio 8:2, respectively. We repeated this process ten times to create ten different training and testing sets (i.e., trained and tested ten versions of each model).

A pair of Convolution Recurrent Neural Networks with the architecture proposed in Figure 1 were trained on each "mixed" and "sep" version of the waveform data. The raw waveform is converted to spectrogram form with the 129 frequency features for each time point. To determine how many time points a 45-second-long song has, we performed a scan through all instances of the dataset and converted them into spectrograms first. We took the max-length spectrogram of all the waveform instances and then padded all the other spectrograms with zeros on the end. By this method, the default spectrogram time length for a song was 15,502. However, it would be resource-demanding if we used such a large array of numbers in the neural network (i.e., 129 x 15,502). Therefore, we used a resizing image layer to condense the spectrogram before fitting the features in the convolutions, and the time axis was resized to 2,048 features. For the "mixed" dataset, the spectrogram was processed through one-channel input. For the "sep" dataset, because there are four waveforms corresponding to four separate instruments, each waveform was converted to a separate spectrogram and concatenated in the channel axis. Therefore, although the pair of Convolution Recurrent Neural Networks have different input channel sizes, both their architectures stay the same.

The input features in Figure 1 go through 5 blocks of convolution; each block contains 2 groups of convolution layers followed by a ReLU layer, and there is a max pooling layer at the end of each convolution block. The convolution layer in the first group has a kernel of 5-by-5, whereas that same layer in the second group has a kernel of 1-by-1. This pointwise convolution (Howard et al. 2017) serves as the linear transformation for each pixel with the vector formed by all features/channels/neurons for that pixel position. The number of features/channels/neurons is also noted in the figure. For the last convolution block, the convolution layer in the first group has a 3-by-3 kernel instead. After the last max pooling layer, the feature vector is flattened and transposed (permute layer) to fit in the 3 LSTM layers with 256, 128, and 64 neurons respectively with Tanh activation. The output of the LSTM layers is then fitted into 3 Dense layers with 512, 256, and 64 neurons respectively with ReLU activation before going through the last Dense layer that has 4 outputs (mean and standard deviation of valence and arousal) with no activation (linear transformation). The model has a total of 5,966,116 parameters. We used the Adam optimizer (Kingma and Ba 2015) to update the gradient in the network. Each Convolution Recurrent Neural model was trained for 77 steps per epoch for four epochs. We also recorded the prediction error with the Mean Square Error function for every seven steps, and the loss pattern can be observed in Figures 2 and 3.

For the Random Forest (RF), we used the Random Forest Regressor supported by the Scikit-Learn library. Our RF model had 100 trees in total, and we used Mean Squared Error as the loss function. In order to ensure all leaves are pure, we did not set any limitation on the depth of trees—the trees kept expanding until all leaves are pure or had less than 2 samples, which is also the min_samples_split hyperparameter when the minimum number of samples per leaf was set to 1. Furthermore, to utilize all features extracted,

Iteration	E_{mixed}	E_{sep}	M_{mixed}	M_{sep}	$W_{\text{mixed, sep}}$
1	0.948	1.783	0.526	1.423	0.000
2	1.075	0.992	0.570	0.753	0.512
3	0.890	1.073	0.590	0.686	0.033
4	0.846	1.037	0.604	0.804	0.002
5	0.775	0.832	0.526	0.626	0.004
6	0.703	0.945	0.530	0.647	0.000
7	0.829	1.666	0.567	1.130	0.000
8	1.012	0.994	0.644	0.589	0.190
9	1.195	1.968	0.740	1.061	0.000
10	1.107	1.272	0.712	0.836	0.006

Table 1: Statistics of all KL Divergence when using Convolution Recurrent Neural Network to evaluate mixed instrument dataset and separate instrument dataset.

Iteration	E_{mixed}	E_{sep}	M_{mixed}	M_{sep}	$W_{\text{mixed, sep}}$
1	2.172	2.035	1.336	1.151	0.020
2	1.961	1.875	1.373	1.302	0.083
3	2.369	2.270	1.530	1.539	0.466
4	2.097	1.882	1.180	1.076	0.018
5	2.476	2.385	1.399	1.332	0.203
6	2.371	2.247	1.465	1.288	0.128
7	2.120	2.199	1.230	1.237	0.549
8	2.376	2.350	1.261	1.272	0.714
9	2.135	2.120	1.239	1.361	0.862
10	2.440	2.446	1.129	1.101	0.932

Table 2: Statistics of all KL Divergence when using Random Forest to evaluate mixed instrument dataset and separate instrument dataset.

we set the hyper-parameter Max Feature to None or 1.0 as default.

In terms of training the RF models, we also had a pair of models for different inputs (one for "mixed" and one for "sep"). For the original "mixed" data, we used all 6,373 features extracted with OpenSMILE to fully demonstrate PMEmo's capability in MER tasks (Zhang et al. 2018). Therefore, the input was a two-dimensional matrix with the shape of (num_samp x 6,373), which is the number of samples-by-the number of features. Similarly, the input for the "sep" data was also in two dimensions with the number of samples-by-the number of features, respectively. However, in order to help the RF model learn relationships between each instrument that might affect the perceived emotion, we concatenated all features from the four separated waveforms into one matrix. So, the input data for this model is $6,373 * 4$ features, shaping the input as (num_samp x 25,492). The output of each RF model is similar to our Convolution Recurrent Neural Network models, which is a two-dimensional array with shape (out_samp x 4): arousal mean, valence mean, arousal standard deviation, and valence standard deviation.

We performed model training on each iteration and validated on the respective testing set of that iteration. Our last step was to calculate the KL Divergence using predicted and ground truth statistics as in Equation 1 for both Convolution Recurrent Neural Network and Random Forest MER models using either "mixed" or "sep." For each iteration, we also plotted the histogram of KL Divergences computed from the result prediction of model pairs ("mixed" or "sep" datasets) for both Convolution Recurrent Neural Network (ten cross-validation iterations) and Random Forest (ten cross-validation iterations). To plot these non-discrete values in a histogram, we divided each particular histogram's KL Divergence domain into thirty bins. We visualized the results of all ten iterations per approach as shown in Figure 4. After that, we computed the mean and median KL Divergence for each iteration's model trained on different datasets. We also computed the Wilcoxon test p-values.

5 Results

We achieved convergence in training all the Convolution Recurrent Neural Networks and computed the Mean Square Error on the testing set for each iteration in 10-iteration cross-validation. The mean loss for ten iterations converged at 0.04 and 0.05 for the model trained on the "mixed" dataset and the model trained on the "sep" dataset, respectively. The convergence of the mean loss can be seen in Figures 2 and 3. Similarly, for the Random Forest, the two models converged at 0.03 for both the models trained on the "mixed" and "sep" datasets. This shows that our models identified a pattern under the loss criteria that maps music waveforms to emotion distributions.

In Tables 1 and 2, we denote "Iteration" as a full experiment run with randomized training and testing set (Section 4) following the Monte Carlo Cross-Validation (Xu and Liang 2001). We denote E_{dataset} as the expected value and M_{dataset} as the median value when evaluating the corresponding dataset's distribution of KL Divergences. We also denote W as the p-value of the Wilcoxon Signed Rank Test.

As can be seen from the distributions in Figure 4 and the mean of KL Divergences in Table 1 for Convolution Recurrent Neural Network, the model trained on the "mixed" dataset has a higher frequency of low KL Divergences than the model trained on the "sep" dataset. We notice that almost all the mean and median KL Divergences for the models trained on "mixed" dataset are lower than those for the models trained with "sep" dataset. The probability that the MER models for the "mixed" and "sep" datasets have the same KL Divergences from the ground truth is less than 1% in seven of the ten iterations, but two of the iterations have a greater probability that the differences in KL Divergences could be a consequence of random chance.

On the other hand, in the Random Forest statistical results (Table 2), most of the Wilcoxon p-values are great enough to imply that the differences in KL Divergences between the "mixed" and "sep" models from the ground truth might be random chance. Figure 4 presents further evidence as the two distributions look very similar in most iterations. This means that there is little-to-no difference between the Random Forest models (using either the "mixed" or "sep" dataset) when predicting music emotion.

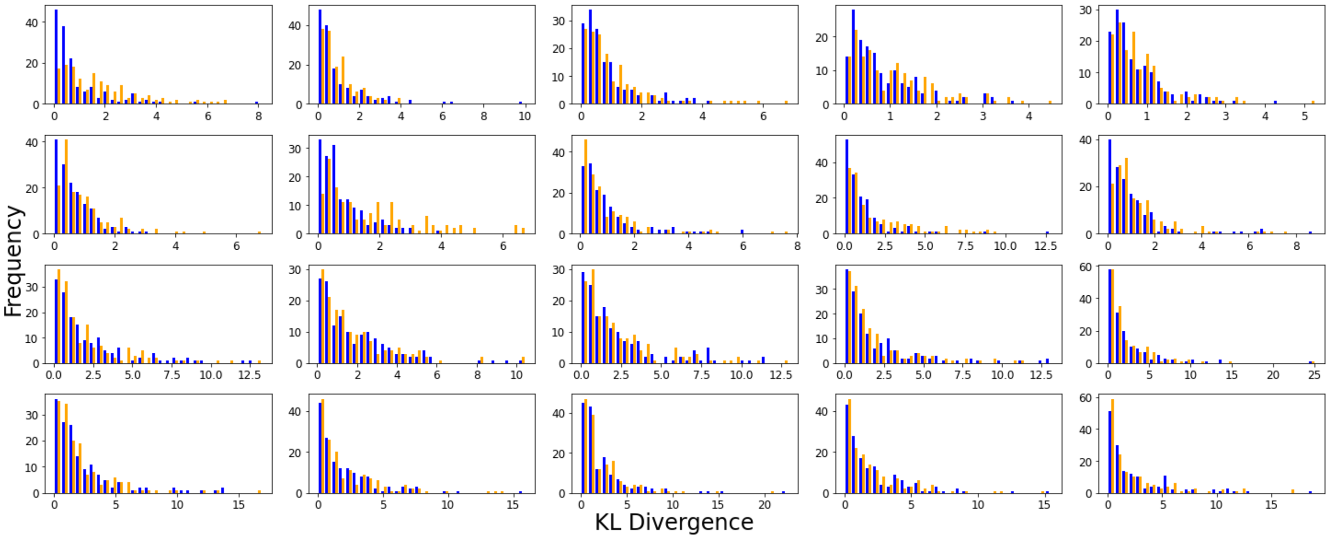

Figure 4: 10-iteration cross-validation of KL Divergence Distribution Comparison for Convolution Recurrent Neural Network (first two rows) and Random Forest (last two rows). Blue shows the KL Divergence distribution of model evaluation on the mixed instrument dataset. Orange shows the KL Divergence distribution of model evaluation on the separated instrument dataset. The x-axis shows the KL Divergence, and the y-axis shows the frequency of that bin. The KL Divergence domain was divided into thirty bins.

6 Discussion

Based on the results of the Wilcoxon Signed Rank Tests, we have empirical evidence that for the Convolution Recurrent Neural Network, it is more accurate to predict music emotion with the mixed instrumental waveform rather than considering each instrument separately. On the other hand, we have empirical evidence that for the Random Forest, neither representation of the waveform can predict music emotion more accurately. Despite the psychological evidence that people account for timbre when perceiving emotion in music, we were unable to implement this phenomenon within a machine learning framework using our approach.

The design of the "sep" dataset in this research context can be considered naïve because it only has three types of instruments (vocal, drum, and bass) and an "others" category for all other timbres. This might not be sufficient for the music in the PMEmo dataset. In particular, the "other" waveforms are combinations of multiple instruments that cannot be extracted from the mixed waveform. The experiments in this paper took the form of training our "sep" model with a few separated music sources, but one of the waveforms was still a mixed waveform. Therefore, we hypothesize for future work that the "sep" dataset did not fully represent the actual "instrumentally separated" dataset distribution as we meant for it to be. The lack of a robust dataset might leave our outcome inconclusive, but it has laid the groundwork for further avenues of investigation.

7 Conclusion

Inspired by psychology research about how people perceive emotion in music, we hypothesized that machine learning models that perceive music emotion will have better performance when we use separated instrumental sources/waveforms. We proposed two machine learning algorithms to test this hypothesis: Convolution Recurrent Neural Network and Random Forest. In order to compare the performance and determine which model was better in predicting music emotion, we introduced the use of KL Divergence to assess statistical representations for the uncertainty over music emotion regression. Statistical analysis of the KL Divergence results presented evidence rejecting our hypothesis.

This work provides a lot of interesting questions for future research, and there are several directions that we can go. First, further work regarding domain adaptation or dataset distribution shift detection for the music datasets could be done because the pretrained Wave-U-Net model was trained on MUSDB18 (Stoller, Ewert, and Dixon 2018), but was evaluated on the PMEmo dataset (Section 3). Another direction is to consider other machine learning algorithms, features, and/or neural network architectures to better observe the patterns related to the statistical distributions for emotion annotations. Another direction is to introspect on our results, inspecting every song that has a large KL Divergence in one of the models; there might be something to learn about when the model(s) each perform well or fail. Each of these methods could lead to a further understanding of how to make automated music emotion recognition better consider the factors that humans use for the task.

Acknowledgments

The authors would like to thank the anonymous reviewers for their feedback that improved the quality of this paper.

References

Aljanaki, A.; Yang, Y.-H.; and Soleymani, M. 2017. Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3): 1–22.

Badshah, A. M.; Ahmad, J.; Rahim, N.; and Baik, S. W. 2017. Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network. In *2017 International Conference on Platform Technology and Service (PlatCon)*, 1–5.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45: 5–32.

Chowdhury, S.; Vall, A.; Haunschmid, V.; and Widmer, G. 2019. Towards Explainable Music Emotion Recognition: The Route via Mid-level Features. *ArXiv*, abs/1907.03572.

Cutler, A.; Cutler, D.; and Stevens, J. 2011. *Random Forests*, volume 45, 157–176. Springer. ISBN 978-1-4419-9325-0.

Dong, Y.; Yang, X.; Zhao, X.; and Li, J. 2019. Bidirectional Convolutional Recurrent Sparse Network (BCRSN): An Efficient Model for Music Emotion Recognition. *IEEE Transactions on Multimedia*, 21(12): 3150–3163.

Eyben, F.; Wöllmer, M.; and Schuller, B. 2010. Opensmile: The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, 1459–1462. New York, NY, USA: Association for Computing Machinery. ISBN 9781605589336.

Gabrielsson, A. 2002. Emotion Perceived and Emotion Felt: Same or Different? *Musicae Scientiae*, 5: 123–147.

Hailstone, J. C.; Omar, R.; Henley, S. M.; Frost, C.; Kenward, M. G.; and Warren, J. D. 2009. It's not what you play, it's how you play it: Timbre affects perception of emotion in music. *Quarterly journal of experimental psychology*, 62(11): 2141–2155.

Han, D.; Kong, Y.; Han, J.; and Wang, G. 2022. A survey of music emotion recognition. *Frontiers of Computer Science*, 16(6): 1–11.

Hizlisoy, S.; Yildirim, S.; and Tufekci, Z. 2021. Music emotion recognition using convolutional long short term memory deep neural networks. *Engineering Science and Technology, an International Journal*, 24(3): 760–767.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-term Memory. *Neural computation*, 9: 1735–80.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv*, abs/1704.04861.

Izard, C. 2008. Emotion Theory and Research: Highlights, Unanswered Questions, and Emerging Issues. *Annual review of psychology*, 60: 1–25.

Juslin, P. 2013. What does music express? Basic emotions and beyond. *Frontiers in Psychology*, 4.

Juslin, P. N.; and Laukka, P. 2004. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of new music research*, 33(3): 217–238.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.-S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; and Patras, I. 2012. DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, 3(1): 18–31.

Kullback, S.; and Leibler, R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79 – 86.

Lin, Y.-C.; Yang, y.-h.; and Chen, H. 2011. Exploiting online music tags for music emotion classification. *ACM Transactions on Multimedia Computing, Communications, and Applications - TOMCCAP*, 7: 1–16.

Liu, X.; Chen, Q.; Wu, X.; Liu, Y.; and Liu, Y. 2017. CNN based music emotion classification. *ArXiv*, abs/1704.05665.

Liutkus, A.; Fitzgerald, D.; and Rafii, Z. 2015. Scalable audio separation with light Kernel Additive Modelling. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 76–80.

Malik, M.; Adavanne, S.; Drossos, K.; Virtanen, T.; Ticha, D.; and Jarina, R. 2017. Stacked Convolutional and Recurrent Neural Networks for Music Emotion Recognition.

Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.-Y.; and Sainath, T. 2019. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2): 206–219.

Rafii, Z.; Liutkus, A.; Stöter, F.-R.; Mimilakis, S. I.; and Bittner, R. 2017. The MUSDB18 corpus for music separation. https://doi.org/10.5281/zenodo.1117372. Accessed: 2022-12-13.

Refaeilzadeh, P.; Tang, L.; and Liu, H. 2009. Cross-Validation. In Liu, L.; and Özsu, M. T., eds., *Encyclopedia of Database Systems*, 532–538. Springer US. ISBN 978-0-387-39940-9.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature*, 323: 533–536.

Russell, J. 1980. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, 39: 1161–1178.

Schuller, B.; Steidl, S.; Batliner, A.; Hirschberg, J.; Burgoon, J.; Baird, A.; Elkins, A.; Zhang, Y.; Coutinho, E.; and Evanini, K. 2016. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity and Native Language. In *Proceedings of Interspeech 2016*, 2001–2005.

Soch, J.; of Statistical Proofs, T. B.; Faulkenberry, T. J.; Petrykowski, K.; and Allefeld, C. 2020. *StatProofBook/StatProofBook.github.io: StatProofBook 2020*. Zenodo.

Song, Y.; Dixon, S.; Pearce, M.; and Halpern, A. 2016. Perceived and Induced Emotion Responses to Popular Music: Categorical and Dimensional Models. *Music Perception: An Interdisciplinary Journal*, 33: 472–492.

Stoller, D.; Ewert, S.; and Dixon, S. 2018. Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation. *ArXiv*, abs/1806.03185.

Vempala, N. N.; and Russo, F. A. 2018. Modeling Music Emotion Judgments Using Machine Learning Methods. *Frontiers in Psychology*, 8.

Xu, Q.-S.; and Liang, Y.-Z. 2001. Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1): 1–11.

Yang, P.-T.; Kuang, S.-M.; Wu, C.-C.; and Hsu, J.-L. 2020. Predicting Music Emotion by Using Convolutional Neural Network. In *HCI in Business, Government and Organizations: 7th International Conference, HCIBGO 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*, 266–275. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-50340-6.

Zhang, K.; Zhang, H.; Li, S.; Yang, C.; and Sun, L. 2018. The PMEmo Dataset for Music Emotion Recognition. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, 135–142. Yokohama, Japan: Association for Computing Machinery. ISBN 9781450350464.