Music emotion recognition using recurrent neural networks and pretrained models

Jacek Grekow¹

Received: 12 December 2020 / Revised: 11 June 2021 / Accepted: 12 July 2021 /
Published online: 8 August 2021
© The Author(s) 2021

Abstract

The article presents conducted experiments using recurrent neural networks for emotion detection in musical segments. Trained regression models were used to predict the continuous values of emotions on the axes of Russell's circumplex model. A process of audio feature extraction and creating sequential data for learning networks with long short-term memory (LSTM) units is presented. Models were implemented using the WekaDeeplearning4j package and a number of experiments were carried out with data with different sets of features and varying segmentation. The usefulness of dividing the data into sequences as well as the point of using recurrent networks to recognize emotions in music, the results of which have even exceeded the SVM algorithm for regression, were demonstrated. The author analyzed the effect of the network structure and the set of used features on the results of the regressors recognizing values on two axes of the emotion model: arousal and valence. Finally, the use of a pretrained model for processing audio features and training a recurrent network with new sequences of features is presented.

Keywords Emotion detection · Audio features · Sequential data · Recurrent neural networks

1 Introduction

Music is an organization of sounds over time, and one of its more important functions is the transmission of emotions. The music created by a composer is ultimately listened to by a listener. The carriers of emotions are sounds distributed over time, their quantity, pitch, timbre, loudness, and their mutual relations. These sounds in music terminology are described by melody, instruments, dynamics, rhythm, and harmony. Before a person notices the emotions in music, he/she must have some time to analyze the listened to fragment (Bachorik et al., 2009); depending on the changes in melody, timbre, dynamics, rhythm, or harmony, we can notice different emotions, such as happy, angry, sad, or relaxed.

✉ Jacek Grekow
j.grekow@pb.edu.pl

¹ Faculty of Computer Science, Bialystok University of Technology,
Wiejska 45A, Bialystok 15-351, Poland

The aim of this paper was to imitate the time-related perception of emotions in music by humans through the construction of an automatic emotion detection system using recurrent neural networks (RNN). Just as the human brain is “fed” with subsequent sound information over time, on the basis of which it perceives the emotions in music, similarly, the neural network downloads subsequent information vectors in subsequent time steps to predict the emotion value of the analyzed musical fragment.

2 Related work

Division into categorical and dimensional approach can be found in papers devoted to music emotion recognition (MER). In the categorical approach, a number of emotional categories (adjectives) are used for labeling music excerpts (Lu et al., 2006; Grekow, 2015; Patra et al., 2017). In the dimensional approach, emotion is described using dimensional space, like the 2D model proposed by Russell (1980), where the dimensions are represented by arousal and valence (Weninger et al., 2014; Coutinho et al., 2015; Grekow, 2016; Delbouys et al., 2018; Grekow, 2018b).

MER task can also be divided into static or dynamic, where static MER detects emotions in a relatively long section of music of 15–60 s (Delbouys et al., 2018; Patra et al., 2018; Chowdhury et al., 2019), and dynamic MER examines changes in emotions over the course of a composition, for example, every 0.5 or 1 s. Dynamic MER task was conducted by MediaEval Benchmarking Initiative for Multimedia Evaluation, the results of which were presented by Aljanaki et al. (2017).

A comprehensive review of the current emotionally-relevant computational audio features used in MER was presented by Panda et al. (2020). They show the relations between eight musical dimensions (melody, harmony, rhythm, dynamics, timbre, expressivity, texture, and form) and specific emotions.

Long-short term memory recurrent neural networks were used in dynamic MER task by Coutinho et al. (2015). Low-level acoustic descriptors extracted using openSMILE and psychoacoustic features extracted with the MIR Toolbox were used as input data. A multivariate regression performed by deep recurrent neural networks was used to model the time-varying emotions (arousal, valence) of a musical piece (Weninger et al., 2014). In this work, a set of acoustic features extracted from segments of 1 s length were used.

Delbouys et al. (2018) used mel-spectrogram from audio and embedded lyrics as input vectors to the convolutional and LSTM networks. Chowdhury et al. (2019) used VGG-style convolutional neural networks to detect 8 emotional characteristics (happy, sad, tender, fearful, angry, valence, energy, tension). For network training perceptual mid-level features (melodiousness, articulation, rhythmic stability, rhythmic complexity, dissonance, tonal stability, modality) were used, and spectrograms from audio signals were used as input vector for neural networks. Deep signal processing architectures and feature learning that can be used in content-based music informatics retrieval (MIR) challenges were presented by Humphrey et al. (2012).

The use of pretrained models in MIR classification tasks was presented in (Hamel et al., 2013; Oord et al., 2014). (Choi et al., 2017) used a pretrained convolutional neural network for music classification and regression tasks. A pretrained on mel-spectrograms model is used as a feature extractor in six music information retrieval and audio-related tasks. The proposed approach uses features from every convolutional layer after applying average pooling to reduce their feature map sizes.

What distinguishes this work from others is that it uses a different segment length (6 s) than the standard static MER, as well as proposes a method of preparing data for recurrent neural networks, which it tests with various low and mid-level features. Due to the fact that the studied segment is relatively short, a solution of using a sliding window also allows to study changes in emotions throughout the entire composition, i.e. similar to dynamic MER. This article is an extension of a conference paper (Grekow, 2020) where the problem was preliminarily presented. In the presented article, the emotion detection method has been expanded to include the use of a pretrained model for processing audio features.

The rest of this paper is organized as follows. Section 3 describes the music data set and the emotion model used in the conducted experiments. Section 4 presents the tools used for feature extraction and preparation of data before building the models. Section 5 describes the details of the built recurrent neural networks. Section 6 presents the results obtained while building the models using the features obtained from audio tools. The use of pretrained models as feature extraction in connection with the recurrent neural network is described in Section 7. Finally, Section 8 summarizes the main findings.

3 Music data

A well-prepared database of learning examples affects the results and the correctness of the created models predicting emotions. The advantages of the obtained database are well-distributed examples on the emotion plane as well as congruity between the music experts' annotations. The data set consisted of 324 six-second fragments of different genres of music: classical, jazz, blues, country, disco, hip-hop, metal, pop, reggae, and rock. The tracks were all 22050 Hz mono 16-bit audio files in .wav format. The training data were taken from the publicly available GTZAN¹ data collection (Tzanetakis & Cook, 2002). After the selection of samples, the author shortened them to the first 6 seconds, which is the shortest possible length at which experts could detect emotions for a given segment. Bachorik et al. (2009) investigated the length of time required for participants to initiate emotional responses to musical samples. On average, participants with varying musical training required 8 seconds of music before initiating emotional judgments. In our experiment, we used five music experts, thus it was decided that the samples will be shortened to 6 seconds.

Data annotation was done by five music experts with a university musical education. The musical education of the experts, people who deal with the creation and analysis of emotions in music on a daily basis, allows us to trust the quality of their annotations. Each annotator annotated all records in the data set - 324 six-second fragments. Each music expert had heard all the examples in the database. As a result during the annotation each annotator was able to see all the shades of emotions in music, which is not always the case in databases with the emotions determined. This had a positive effect on the quality of the received data, which was emphasized by Aljanaki et al. (2017).

During annotation of music samples, we used the two-dimensional arousal-valence Russell's model (Fig. 1) to measure emotions in music, which consists of two independent dimensions of arousal (vertical axis) and valence (horizontal axis). Each music expert making annotations after listening to a music sample had to specify values on the arousal and valence axes in a range from -10 to 10.

¹<https://www.tensorflow.org/datasets/catalog/gtzan>

Fig. 1 Russell's circumplex model (Russell, 1980)

Fig. 2 Data set on A-V emotion plane

Table 1 Amount of examples in quarters on the A-V emotion plane

Quarter		Amount
Abbreviation	Arousal-Valence	of examples
Q1	high-high	93
Q2	high-low	70
Q3	low-low	80
Q4	low-high	81

Value determination on the arousal-valence axes (A-V) was clear with a designation of a point on the A-V plane corresponding to the musical fragment. The data collected from the five music experts were averaged. Figure 2 presents the annotation results of a data set with A-V values. The amount of examples obtained in the quarters on the A-V emotion plane is presented in Table 1.

A well-prepared database, i.e. one suitable for independent regressors predicting valence and arousal, should contain examples where the values of valence and arousal are not correlated. To check if valence and arousal dimensions are correlated in our music data, the Pearson correlation coefficient was used. The obtained value of $r = -0.03$ (i.e. close to zero) indicates that arousal and valence values are not correlated and the music data are a well spread in the quarters on the A-V emotion plane.

All examples in the database were marked by five music experts and their annotations had good agreement levels. A good level of mutual consistency was achieved, represented by Cronbach's α calculated for the annotations of arousal ($\alpha = 0.98$) and valence ($\alpha = 0.90$). We can see that the experts' annotations for the arousal value show greater agreement than for the valence value, which is in line with the natural perception of emotions by humans (Aljanaki et al., 2017). Details on creating the music data were presented in a previous paper (Grekow, 2018a). The collected music data set is available on the web site.²

4 Audio feature extraction

4.1 Tools for feature extraction

For feature extraction, tools for audio analysis and audio-based music information retrieval, Essentia (Bogdanov et al., 2013) and Marsyas (Tzanetakis & Cook, 2000), were used. Marsyas software, written by George Tzanetakis, has the ability to analyze music files and to output the extracted features. The tool enables the extraction of the following features: Zero Crossings, Spectral Centroid, Spectral Flux, Spectral Rolloff, Mel-Frequency Cepstral Coefficients (mfcc), and chroma features - 31 features in total. For each of these basic features, Marsyas calculates four statistic features (mean, variance and higher-order statistics over larger time windows). The feature vector length obtained from Marsyas was 124.

Essentia is an open-source library, created at the Music Technology Group, Universitat Pompeu Fabra, Barcelona. In the Essentia package, we can find a number of executable extractors computing music descriptors for an audio track: spectral, time-domain, rhythmic,

²<https://github.com/grekowj/musemodataset>

and tonal descriptors. Extracted features by Essentia are divided into three groups: low-level, rhythm, and tonal features. A full list of features is available on the web site.³ Essentia also calculates many statistics over values collected in array: the mean, geometric mean, power mean, median of an array, all its moments up to the 5th-order, its energy, and the root mean square (RMS). The feature vector length obtained from Essentia was 529.

4.2 Preparing data for RNN

Recurrent neural networks process sequential data and find relationships between the input data sequences and the expected output value. To be able to train the recurrent neural network, it is necessary to enter sequences of the feature vectors. In this paper, to extract correlations with time in the studied music fragments, they were segmented into smaller successive sections. The process of dividing a fragment of music (6 s) into smaller segments of a certain length t (1, 2 or 3 s) and overlap (0 or 50%) is shown in Fig. 3. To split the wav file, the *sfplay.exe* tool from Marsyas toolkit was used. From the created smaller segments of music, feature vectors were extracted, which were used to build a sequence of learning vectors for the neural network. A program was written that allows to select the segmentation option for a music fragment, performs feature extraction, and prepares data to be loaded to a neural network.

5 Recurrent neural networks

Long short-term memory (LSTM) units, which were defined in Gers et al. (2000), were used to build recurrent networks. LSTM units are special kinds of memory blocks that solve the vanishing gradient problem occurring with simple RNN units. Each LSTM unit consists of a self-connected memory cell and three multiplicative regulators - input, output, and forget gates. Gates provide LSTM cells with write, read, and reset operations, which allows the LSTM unit to store and access information contained in a data sequence that corresponds to data distributed over time. The weights of connections in LSTM units need to be learned during training.

5.1 Implementation of RNN

The WekaDeeplearning4j package (Lang et al., 2019), which was included with the Weka program (Hall et al., 2009), was used to conduct the experiments with recurrent neural networks. This package makes deep learning accessible through a graphical user interface. The WekaDeeplearning4j module is based on Deeplearning4j,⁴ which is a widely used open-source machine learning workbench implemented in Java. Weka with WekaDeeplearning4j package enables users to perform experiments by loading data in the Attribute-Relation File Format (ARFF), configuring a neural network, and running the experiment.

To predict emotions in music files, a neural network was proposed with the structure shown in Fig. 4. Input data were given to the network in the form of a sequence set of feature vectors, and then processed by a layer consisting of LSTM units (LSTM1–LSTMn). The

³http://essentia.upf.edu/documentation/algorithms_reference.html

⁴<https://deeplearning4j.org>

Fig. 3 Creating training data sequences for RNN

next layer built of densely connected neurons (1– n) converted the signals received from the LSTM layer and created an output signal.

5.2 ARFF data for RNN

The Weka program allows to load learning data in the ARFF format. During training, the recurrent neural network from WekaDeeplearning4j package needs sequential data and output training values. The prepared data are slightly different from typical ARFF data because they contain a *relational* attribute that specifies the set of features found in each step of the sequence (code below). The definition of the feature set ends with the *@end* keyword, which in our case refers to *@attribute bag relational*. The data at each time step is separated by $\backslash n$. In the data section, the entire sequence of one example is written on one line enclosed in quotation marks (") and terminated with the *output* value. To prepare data for the neural network implemented using the WekaDeeplearning4j package, the author wrote a script that converts vectors obtained during feature extraction into sequences saved in one

Fig. 4 Recurrent neural network architecture

ARFF format file. Below is an example of training data containing attribute definitions and two data sets with a 3-step sequence, and the numeric *output* value (value of arousal).

```
@relation Arousal_Sequential_Data
@attribute bag relational
    @attribute Mean_MFCC0 numeric
    @attribute Mean_MFCC1 numeric
    @attribute Mean_MFCC2 numeric
    ...
    @attribute feature_no_124 numeric
@end bag
@attribute output numeric

@data
"-48.145309,5.329454,-0.679031, ... 1.027434,\n
-50.730044,6.186828,0.431127, ... 0.435338,\n
-47.743233,6.319406,-0.482212,... 0.505049,\n",0.29
"-55.545411,6.869730,1.128843, ... 0.106391,\n
-55.178950,9.128733,0.437831, ... -0.151600,\n
-52.558833,7.650667,-0.268933, ... 0.136838,\n",-0.2875
...
```

[Example of learning data with a 3-step sequence]

5.3 Parameters of the RNN

The structure of the neural network was built once with one LSTM layer, once with two layers, and with different amounts of LSTM units (124, 248). A tanh activation function was used for LSTM units. For our regression task (prediction of continuous values of arousal and valence), the identity activation function for a dense layer was used, in conjunction with the mean squared error loss function. For weight initialization, the Xavier method was used and the Nesterov updater helped to optimize the learning rate. The network was trained with 100 epochs and to avoid overfitting an early stopping strategy was used. The training process was stopped as soon as the loss did not improve anymore for 10 epochs. The loss was evaluated on a validation set (20% of the training data).

6 Experiments and results

During the conducted experiments, regressors for predicting arousal and valence were built. As baseline for comparing the results of the obtained regressors a simple linear regression model (lr) was chosen. The data were also tested with second baseline SMOREg algorithm with polynomial kernel, which is an implementation of the support vector machine for regression. The author also tested the usefulness of SMOREg on the same database in previous papers (Grekow, 2016; 2017). In our experiments, both baseline algorithms (SMOREg, lr) were tested on the same music fragments as the neural networks but on non-segmented fragments. These two algorithms were trained using data obtained from the whole (6 s) music samples.

The regression algorithms were evaluated using the 10-fold cross validation technique (CV-10). The coefficient of determination (R^2) and mean absolute error (MAE) were used to assess model efficiency. Before constructing regressors arousal and valence annotations were scaled between $[-0.5, 0.5]$. Before providing input data to the neural network, the data was standardized to zero mean and unit variance.

Regressors were built using RNN (RnnSequenceClassifier (Lang et al., 2019)) and were tested in 4 variants:

- RNN1 - 1 layer x 124 LSTM units;
- RNN2 - 2 layers x 124 LSTM units each;
- RNN3 - 1 layer x 248 LSTM units;
- RNN4 - 2 layers x 248 LSTM units each.

6.1 RNN with Marsyas features

During the testing of RNN efficiency, features obtained from Marsyas tool were divided into 3 sets:

- All Marsyas features (124);
- Mfcc features - 13 Mel Frequency Cepstral Coefficients x 4 statistic (52);
- Chroma features - (12 pitch class features + minimum and average for A pitch class) x 4 statistic (54).

Tables 2 and 3 present the coefficient of determination (R^2) and mean absolute error (MAE) obtained during building regressors using chroma and mfcc features. The best results for each regressor type (arousal, valence) are marked in bold. From the obtained results, we can see that the usefulness of the chroma features is small compared with the

Table 2 Results obtained for chroma features

Model	Arousal		Valence	
	R^2	MAE	R^2	MAE
Linear regression	0.40	0.17	0.02	0.18
SMOREg	0.43	0.16	0.04	0.17
RNN1 (124 LSTM)	0.43	0.17	0.02	0.18
RNN2 (124, 124 LSTM)	0.48	0.15	0.03	0.17
RNN3 (248 LSTM)	0.41	0.17	0.02	0.18
RNN4 (248, 248 LSTM)	0.46	0.16	0.02	0.18

Table 3 Results obtained for mfcc features

Model	Arousal		Valence	
	R^2	MAE	R^2	MAE
Linear regression	0.61	0.13	0.07	0.17
SMOreg	0.60	0.13	0.10	0.17
RNN1 (124 LSTM)	0.58	0.14	0.14	0.17
RNN2 (124, 124 LSTM)	0.66	0.12	0.11	0.16
RNN3 (248 LSTM)	0.61	0.13	0.13	0.16
RNN4 (248, 248 LSTM)	0.64	0.12	0.14	0.15

mfcc features. The results for mfcc features far outweigh those for chroma features. A significant improvement can be seen in the case of R^2 for valence.

Table 4 presents the results for all Marsyas features. A simple linear regression model and support vector machine for regression (SMOreg) were outperformed by the RNN models, in two cases RNN2, RNN4 for arousal and valence. The best results were obtained with RNN4 (2 layers x 248 LSTM): $R^2 = 0.67$ and $MAE = 0.12$ for arousal, $R^2 = 0.17$ and $MAE = 0.15$ for valence. We see that RNN with two LSTM layers gives better results for both arousal as well as valence. As expected, the results show that the sequential modeling capabilities of the RNN are useful for this task.

The use of all features gives the best results; however, in the case of arousal, the set of mfcc features gives quite comparable results, similar to the whole set of features ($R^2 = 0.66$ and $MAE = 0.12$, Table 3). The best results were obtained at a segment length of 2 s and without overlap, and those are presented here.

6.2 RNN with Essentia features

Experiments with the features obtained from the Essentia package were also conducted. These features include the mfcc and chroma features, which are also in the Marsyas tool, but also contain many higher-level features such as rhythm or harmony. Table 5 shows the results of the experiments. The experiments were expanded by two networks with an increased number of LSTM units, similar to the number of features in the sequence:

- RNN5 - 1 layer x 529 LSTM units;
- RNN6 - 2 layers x 529 LSTM units each.

Table 4 Results obtained for all Marsyas features

Model	Arousal		Valence	
	R^2	MAE	R^2	MAE
Linear regression	0.56	0.14	0.13	0.17
SMOreg	0.62	0.13	0.15	0.16
RNN1 (124 LSTM)	0.58	0.14	0.12	0.17
RNN2 (124, 124 LSTM)	0.66	0.12	0.16	0.15
RNN3 (248 LSTM)	0.62	0.13	0.12	0.17
RNN4 (248, 248 LSTM)	0.67	0.12	0.17	0.15

Table 5 Results obtained for Essentia features

Model	Arousal		Valence	
	R^2	MAE	R^2	MAE
Linear regression	0.07	0.25	0.06	0.19
SMOreg	0.48	0.18	0.27	0.17
RNN1 (124 LSTM)	0.54	0.14	0.21	0.16
RNN2 (124, 124 LSTM)	0.67	0.12	0.32	0.13
RNN3 (248 LSTM)	0.58	0.14	0.32	0.14
RNN4 (248, 248 LSTM)	0.69	0.11	0.40	0.13
RNN5 (529 LSTM)	0.61	0.13	0.29	0.15
RNN6 (529, 529 LSTM)	0.68	0.12	0.36	0.14

From the obtained results (Table 5) for the Essentia feature set, we can see a significant improvement of the results compared with the baseline algorithms (RNN1-RNN6 for arousal, RNN2-RNN6 for valence). Better features from the Essentia toolkit give better neural network results. The best results were obtained with RNN4: $R^2 = 0.69$ and $MAE = 0.11$ for arousal, $R^2 = 0.40$ and $MAE = 0.13$ for valence. The improvement is also significant for regressors for valence, compared with the results from Marsyas features (Table 4), where the best result was: $R^2 = 0.17$ and $MAE = 0.15$.

In regard to the different numbers of layers and LSTM units, the best results were obtained using the RNN4 network (2 layers x 248 LSTM) for both arousal and valence. Two-layer networks recognized emotions better than one-layer networks.

What is quite interesting in the case of arousal ($R^2 = 0.69$, $MAE = 0.11$), the results are comparable with the results obtained from the Marsyas package ($R^2 = 0.67$, $MAE = 0.12$, Table 4). Mfcc features are quite good for detecting arousal, and adding new features improved the results only slightly. A significant result of these experiments is that features from the Essentia package, like rhythm and tonal features, significantly improved the detection of valence. In the case of arousal, it is not necessary to use such a rich set of features, which is why the model for arousal is not so complex.

7 Using pretrained models as feature extraction

The results obtained in previous experiments with Essentia features were not bad, but one could always find a method that would improve these results. As we have noticed, the set of features describing our data (music files) has a significant impact on the learned model results. The better the features we take, the more satisfactory the results, an example of which was the use of features from Essentia. The efforts presented below to improve the results focused on the data that worked best in previous experiments, that is, the data obtained with Essentia as the audio feature extraction tool (Section 6.2).

A known method used in machine learning is feature selection (Witten et al., 2016), which finds the more suitable feature sets among the output features. The method used in the next experiment was using a pretrained model as a feature extractor, and then training RNN on a new set of features.

Fig. 5 Pretrained model

7.1 Pretrained models

To build the pretrained model, a simple neural network (NN) with a dense layer was used. The pretrained model was taught on a task slightly different than the target task because the training was on non-segmented music fragments, which described the entire length of the music segment (6 s), i.e. not on the same segment lengths as for RNN. During the construction of the pretrained model, use of one dense layer with 248 neurons was tested. The trained NN processed the features obtained from the audio feature extraction tool Essentia into a new set of features, which were the activation values of the dense layer.

The structure of the neural network (Fig. 5) was built once with one dense layer with 248 units. A ReLU activation function was used for neurons. For our regression task, the identity activation function for a dense layer was used in conjunction with the mean squared error loss function. The Xavier method was used for weight initialization and the Adam updater helped to optimize the learning rate. The network was trained with 50 epochs and an early stopping strategy was used to avoid overfitting. The training process was stopped as soon as the loss did not improve anymore for 10 epochs and the loss was evaluated on a validation set (20% of the training data). Due to the fact that regressors were built for two tasks, arousal prediction and valence prediction, separate pretrained models were created for each of them.

7.2 Model construction using a pretrained model

Feature vector transformation and the connection with the RNN were conducted in the Weka program using D14jMlpFilter (Lang et al., 2019). This tool enabled the use of a pretrained model as a feature extractor. Connecting the pretrained model with the RNN is shown in Fig. 6. Activations in the last layer of the pretrained model were used as input data in the RNN. The input data were given to the network in the form of a sequence set of feature vectors. The pretrained model was then used to transform the feature vectors into new feature vectors. Each vector from the input sequence was transformed separately, so that a sequence set of new feature vectors was obtained at the LSTM layer input. The new feature vectors were processed by a layer consisting of LSTM units (LSTM1–LSTMn). The last layer, built of densely connected neurons (1–n), converted the signals received from the LSTM layer and created an output signal. Just as in the previous experiment (Section 6.2),

Fig. 6 Recurrent neural network architecture using the pretrained model

the structure of the neural network was built once with one LSTM layer, once with two layers, and with different amounts of LSTM units (124, 248, 529), which resulted in 6 variants of RNN1–RNN6.

7.3 Results

Table 6 shows the results obtained during the experiments using the pretrained models as feature extraction. We can notice a significant improvement in the coefficient of determination (R^2) and a reduction of the mean absolute error (MAE) in the regressors for both arousal and valence compared with the experiments with the Essentia feature set and without the use of pretrained models (Table 5).

The best results were obtained with RNN2 (2 layers x 124 LSTM) and RNN4 (2 layers x 248 LSTM): $R^2 = 0.73$ and $MAE = 0.11$ for arousal. For valence, the best results were obtained with RNN4, RNN6 (2 layers x 529 LSTM): $R^2 = 0.46$ and $MAE = 0.12$. We

Table 6 Results obtained using pretrained model

Model	Arousal		Valence	
	R^2	MAE	R^2	MAE
RNN1 (124 LSTM)	0.63	0.13	0.35	0.14
RNN2 (124, 124 LSTM)	0.73	0.11	0.42	0.13
RNN3 (248 LSTM)	0.68	0.12	0.36	0.14
RNN4 (248, 248 LSTM)	0.73	0.11	0.46	0.12
RNN5 (529 LSTM)	0.69	0.12	0.38	0.13
RNN6 (529, 529 LSTM)	0.71	0.12	0.46	0.12

can see the advantage of RNN with two LSTM layers over networks with one LSTM layer. As in previous experiments, arousal regressors are more accurate than valence. By using feature vectors obtained from the pretrained model, we obtained 6% relative improvement of R^2 of the best models in the case of arousal, and 15% in the case of the valence regressor (Tables 6 and 5). Thus, it seems a better improvement in the valence regressor than the arousal regressor was obtained. The conducted experiments confirmed the point of using the pretrained model as a way to find even better combinations of features based on audio features for training RNN.

8 Conclusions

This article presents experiments using recurrent neural networks for emotion detection for musical segments. The sequential possibilities of the models turned out to be very useful for this type of task as the obtained results exceeded such algorithms as support vector machine for regression, not to mention the weaker linear regression. In all the built models, the accuracy of arousal prediction exceeded the accuracy of valence prediction. There was more difficulty detecting emotions on the valence axis than arousal. Similar difficulties were noted when music experts were annotating files, which was confirmed during annotation compliance testing.

It is significant that the use of higher-level features (features from Essentia tool) had a very positive effect on the models, especially the accuracy of valence regressors. Interestingly, to predict arousal, even a small set of features (mfcc from Marsyas tool) provided quite good results, similar to those of the large feature set from Essentia. Low-level features, like mfcc, are generally sufficient for predicting arousal.

It appears that the use of pretrained models as feature extraction for the Essentia feature set creates a more favorable set of features that can be used for emotion detection by RNN. The obtained results confirm the positive impact of using feature extraction to create even more useful features. Adding new features, such as melody features, to the audio feature extraction tools in the future would be a way to get even better results in detecting emotions in music files, although it turns out that pretrained models also discover useful features for emotion detection in musical segments.

The shortcomings of using the pretrained models is connected with a more complicated analysis which of the input features were used during extracting features. Extracted features are activations from the last layer of the pretrained model, and to find out which input

features were used to create the new features, one should analyze the weight of the layers in the pretrained model.

The experiment results presented in this paper can be used by building automated systems for music emotion recognition. Such systems are applied in all tasks connected with music file analysis in terms of emotions, such as searching files with a given emotion, tracking the emotional development of soundtracks, and comparing the emotional distribution of musical compositions.

Acknowledgements This research was realized as part of study no. WZ/WI-IIT/2/2020 in the Bialystok University of Technology and financed with funds from the Ministry of Science and Higher Education.

Funding Open access funding provided by Bialystok University of Technology.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aljanaki, A., Yang, Y. H., & Soleymani, M. (2017). Developing a benchmark for emotional analysis of music. *PLoS ONE*, 12(3).
- Bachorik, J., Bangert, M., Loui, P., Larke, K., Berger, J., Rowe, R., & Schlaug, G. (2009). Emotion in motion: Investigating the time-course of emotional judgments of musical stimuli. *Music Perception*, 26, 355–364.
- Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & Serra, X. (2013). ESSENTIA: An audio analysis library for music information retrieval. In *Proceedings of the 14th international society for music information retrieval conference* (pp. 493–498).
- Choi, K., Fazekas, G., Sandler, M. B., & Cho, K. (2017). Transfer learning for music classification and regression tasks. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.) *Proceedings of the 18th international society for music information retrieval conference, ISMIR 2017, Suzhou, China, October 23–27, 2017* (pp. 141–149).
- Chowdhury, S., Portabella, A. V., Haunschmid, V., & Widmer, G. (2019). Towards explainable music emotion recognition: The route via mid-level features. In *Proceedings of the 20th international society for music information retrieval conference, ISMIR 2019, Delft, The Netherlands* (pp. 237–243).
- Coutinho, E., Trigeorgis, G., Zafeiriou, S., & Schuller, B. (2015). Automatically estimating emotion in music with deep long-short term memory recurrent neural networks. In *Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany*.
- Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J., & Moussallam, M. (2018). Music mood detection based on audio and lyrics with deep neural net. In *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, Paris, France* (pp. 370–375).
- Gers, F. A., Schmidhuber, J., & Cummins, F. A. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12, 2451–2471.
- Grekow, J. (2015). Audio features dedicated to the detection of four basic emotions. In *Computer information systems and industrial management: 14th IFIP TC 8 international conference, CISIM 2015, warsaw, poland, september 24–26, 2015, Proceedings* (pp. 583–591). Springer International Publishing.
- Grekow, J. (2016). Music emotion maps in arousal-valence space. In *Computer information systems and industrial management: 15th IFIP TC8 international conference, CISIM 2016, Vilnius, Lithuania, Proceedings* (pp. 697–706). Springer International Publishing.
- Grekow, J. (2017). Audio features dedicated to the detection of arousal and valence in music recordings. In *2017 IEEE international conference on innovations in intelligent systems and applications (INISTA)* (pp. 40–44). IEEE. <https://doi.org/10.1109/INISTA.2017.8001129>.

- Grekow, J. (2018a). Human annotation. In *From content-based music emotion recognition to emotion maps of musical pieces* (pp. 13–24). Cham: Springer International Publishing.
- Grekow, J. (2018b). Musical performance analysis in terms of emotions it evokes. *Journal of Intelligent Information Systems*, 51(2), 415–437. <https://doi.org/10.1007/s10844-018-0510-y>.
- Grekow, J. (2020). Static music emotion recognition using recurrent neural networks. In: D. Helic, G. Leitner, M. Stettinger, A. Felfernig, ZW. Ras (Eds.) *Foundations of Intelligent Systems - 25th International Symposium, ISMIS 2020, Graz, Austria, September 23-25, 2020, Proceedings, Springer, Lecture Notes in Computer Science, vol 12117*, pp 150–160.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explor Newsl*, 11(1), 10–18.
- Hamel, P., Davies, M. E. P., Yoshii, K., & Goto, M. (2013). Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity. In A. Jr. de Souza Britto, F. Gouyon, & S. Dixon (Eds.) *Proceedings of the 14th international society for music information retrieval conference, ISMIR 2013, Curitiba, Brazil, November 4-8 2013* (pp. 9–14).
- Humphrey, E. J., Bello, J. P., & LeCun, Y. (2012). Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In F. Gouyon, P. Herrera, L. G. Martins, & M. Müller (Eds.) *Proceedings of the 13th international society for music information retrieval conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12 2012* (pp. 403–408).
- Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M., & Frank, E. (2019). WekaDeeplearning4j: A deep learning package for Weka based on Deeplearning4j. *Knowledge-Based Systems*, 178, 48–50.
- Lu, L., Liu, D., & Zhang, H.J. (2006). Automatic mood detection and tracking of music audio signals. *Transactions on Audio, Speech, and Language Processing*, 14(1), 5–18.
- Oord, A., Dieleman, S., & Schrauwen, B. (2014). Transfer learning by supervised pre-training for audio-based music classification. In H. Wang, Y. Yang, & J. H. Lee (Eds.) *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, Taipei, Taiwan, October 27-31 2014* (pp. 29–34).
- Panda, R., Malheiro, R. M., & Paiva, R.P. (2020). Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*. <https://doi.org/10.1109/TAFFC.2020.3032373>.
- Patra, B., Das, D., & Bandyopadhyay, S. (2017). Labeling data and developing supervised framework for hindi music mood analysis. *Journal of Intelligent Information Systems*, 48, 633–651. <https://doi.org/10.1007/s10844-016-0436-1>.
- Patra, B., Das, D., & Bandyopadhyay, S. (2018). Multimodal mood classification of hindi and western songs. *Journal of Intelligent Information Systems*, 51, 579–596. <https://doi.org/10.1007/s10844-018-0497-4>.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6), 1161–1178.
- Tzanetakis, G., & Cook, P. (2000). Marsyas: a framework for audio analysis. *Org Sound*, 4(3), 169–175.
- Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), 293–302. <https://doi.org/10.1109/TSA.2002.800560>.
- Weninger, F., Eyben, F., & Schuller, B. (2014). On-line continuous-time music mood regression with deep recurrent neural networks. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5412–5416).
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C.J. (2016). *Data mining fourth edition: Practical machine learning tools and techniques*, 4th edn. USA: Morgan Kaufmann Publishers Inc.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.