Music Emotion Analysis Using Semantic Embedding Recurrent Neural Networks

Jan Jakubik
Wrocław University of Science and Technology
Department of Computational Intelligence
Wrocław, Poland
Email: jan.jakubik@pwr.edu.pl

Halina Kwaśnicka
Wrocław University of Science and Technology
Department of Computational Intelligence
Wrocław, Poland
Email: halina.kwasnicka@pwr.edu.pl

***Abstract*—The paper presents an original approach to music emotion recognition. We propose to use recurrent neural networks to separate the representation learning process from the classifier, which allows us to use a Support Vector Machine on top of a network to improve the results. We define a suitable loss function that is able to find a feature space in which similarity between vectors representing the music recordings corresponds to the similarity between their annotations. The proposed method was tested for regression and classification using two datasets. The results are presented and discussed.**

Keywords: recurrent neural network; feature learning; music emotion recognition; music information retrieval

I. Introduction

In recent years, we have seen a huge growth of collected data of all types: numerical, text documents, images, and videos. While content-based retrieval of image and text data remains a very active area of research with fast-paced advancements, another type of data that grows fast in size becomes a subject of interest: digital audio collections. The audio domain is typically considered in machine learning for two types of data, speech recordings and musical recordings. In the case of music, the imperfection of producer-supplied tags describing the music available online makes Music Information Retrieval (MIR) research an important area of study. MIR involves such subtasks as feature extraction, indexing, searching, and browsing, for both actual audio and its symbolic representation.

This paper deals with the automatic recognition of emotion induced by music. Music is often referred to as a language of emotion [1] since it can be argued that its main purpose is evoking emotions in people. There is a complex relationship between features such as timbre, harmony, lyrics, etc. and emotions they can induce [2]. However, there is a high degree of subjectivity regarding music emotion, and the extraction of features is a non-trivial problem by itself. The above makes the emotional content of music difficult to approach from the perspective of automatic recognition and classification.

In this paper, we propose a semantic embedding inspired approach to recognizing music emotion using recurrent neural networks. Our research is in the area of representation learning, otherwise known as feature learning. A feature learning method attempts to find a mapping from a low-level feature space which would be hard to directly supply to a classifier, e.g. individual pixels of an image, into a more informative one. Using a recurrent neural network, we define a loss function applicable to music emotion recognition problem, which allows us to separate the feature learning process from the classifier or regressor. We test the approach on two datasets which originally served as a subject of research on feature selection for music emotion recognition. The experiments allow us to confirm the usefulness of our feature learning approach compared to domain-knowledge based high-level features. Selected data sets allow to test the proposed method for two kinds of emotion recognition problems:

- regression, i.e., continuous emotional scales, where each training sample is annotated by a vector of real numbers
- k-class classification, i.e., models of emotion, where each training sample is annotated by a single label.

The paper is organized as follows: Section II summarizes literature related to our subject, concerning both music emotion recognition and feature learning methods. Section III contains formal definitions of recurrent neural networks we use for feature learning. In section IV, we define the loss function that drives the semantic embedding approach. Section V contains the description of two datasets we performed tests on and describes the experiments.

II. Related Work

The most common approach to music emotion recognition as a machine learning task is to use hand-crafted features based on expert domain knowledge with a classification algorithm such as SVM or a regression method such as SVR. Multiple papers have examined the connection between emotion and different types of features describing music, including spectral, rhythmic, melodic and harmonic characteristics of sound [3].

In [4], LeCun et al. suggested that music information retrieval may benefit from examining an alternative approach: feature learning. In past years, convolutional neural networks, deep belief networks [5][6] and simple sparse coding methods [7][8][9] have been employed in music analysis tasks to great success. These methods, working on a low-level representation of sound, try to learn features that would be useful in the classification process.

A. Music Emotion

An additional layer of complexity is added to the emotion recognition task because there is no single system to categorize or measure emotion. Both categorical [10] and continuous space [11] models of emotion exist, and were employed in music emotion recognition research [12] [13].

The most popular model is two-dimensional continuous Valence-Arousal scale. Valence dimension separates positive emotions from negative ones, while Arousal separates high activation from low. E.g., positive valence and positive arousal can be described as "joy", while positive valence and negative arousal as "calmness". However, and ongoing topic of discussion is whether the scale should be completed with a third dimension, such as Power or Tension [14]. It should be noted that Valence-Arousal scale is not a domain-specific model of emotion and is derived from research concerning emotions in general. In [15], emotion recognition was treated as a regression problem with separate models for valence and arousal, achieving 0.76 correlation coefficient between predicted and actual value for arousal and 0.53 for valence.

An example of a domain-specific categorization of emotions, which we employ in this paper, is Geneva Emotional Music Scale (GEMS) [16]. GEMS emotional categories are based on surveys in which participants were asked for terms they use to describe emotions induced by music. It is important to note that the distinction between induced (caused by music) and expressed (intended by the author) emotion was made explicitly in the surveying process. GEMS categories are organized in a 3-level hierarchy in which top level contains generic emotion clusters and bottom level specific nouns, with middle level consisting of 9 general emotional categories: wonder, transcendence, tenderness, nostalgia, calmness, power, joy, tension, and sadness. As GEMS is focused on emotion induced by music during voluntary listening, the scale is biased towards positive emotions and descriptive terms rather than negative ones. In [17] GEMS emotion recognition was treated as a task of predicting community consensus (regressed variable is the fraction of people agreeing that a specific song causes a specific emotion). The authors used a set of popular music features, and proposed an additional category of harmonic features. In experimental evaluation, harmonic features improved the recognition accuracy. In [18], we have shown that a simple autoencoder neural network used as a feature learning mechanism on a spectrogram of the file can achieve comparable results. The correlation between predicted and actual values was, on average between all emotions, equal to 0.47. However, it should be pointed out that the variance between different emotions was rather high.

B. Feature Learning and Semantic Embedding

The concept of feature learning is associated closely with deep learning paradigm. In feature learning process, the goal is to create a representation of data that can be passed to a classifier or a regression algorithm, resulting in better classification accuracy than training on raw data. A standard deep belief network can be viewed as a feature learning architecture that first learns a representation of the data (initial layers), and then passes this representation to the classifier (final layer). Similarly, convolutional neural networks are tied closely to the concept of feature learning.

We want to separate the feature learning process from classification by defining a learning goal for a neural network, in which the desired outcome is a feature space that retains semantic relations between data points. Our approach to feature learning is inspired by the concept of multi-label embedding, which achieved good results in multi-label annotation problems. Sparse multi-label linear embedding was defined in [19] as an optimization problem in which the goal is to find a transformation of the original feature space to a feature space in which relations between data points resemble their relations in annotation space. The mathematical formulation of "relations" between points, in this case, was based on expressing a data point as a sparse combination of other data points.

In the area of deep learning, Deep Semantic Embedding (DSE) was proposed as an approach to document retrieval [20]. DSE learns similarity between documents and queries, both of which can be expressed in the same abstract space (e.g., word occurrence vectors). This is not possible for music information retrieval. A concept of semi-supervised embedding was demonstrated in [21]. In semi-supervised embedding (SSE) approach, the loss function of a neural network was modified to incorporate unannotated data. However, the goal of the learning process is to create a feature space in which training samples of the same class are close to each other, while the training samples of different classes are as far from each other as possible, which makes it similar to our approach. The limitation of SSE is that it requires well-defined classes and is not applicable to continuous dimensional models of emotion. Neither of the approaches is applicable to time series data.

III. Recurrent Neural Networks

A recurrent neural network (RNN)[22] is a type of artificial neural network allowing for time series modeling. Given a series of l input vectors $X = (x_1, x_2, ..., x_l)$, a basic recurrent layer is defined by the equation (1):

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{1}$$

which calculates the activation of a layer at time t given its previous activation h_{t-1} and current input x_t. σ is an activation function, applied to every element of a vector. We will denote logistic sigmoid activation as σ_{sig} and hyperbolic tangent activation as σ_{tanh}. Weight matrices W, U and the bias vector b are learned using backpropagation algorithm, unfolding the recurrent network into a deep network.

Due to successful application in the natural language processing domain, more complex models using gating mechanisms became an area of focus for neural network researchers. In these models, a recurrent layer is replaced by a "unit", consisting of multiple interconnected layers, outputs of which can

be added or multiplied element-wise. In particular, element-wise multiplication of any output with an output of a log-sigmoid layer creates a "gating" mechanism in which the log-sigmoid layer can be interpreted as a gate deciding whether the output passes through (multiplication by 1) or does not (multiplication by 0). One of the most popular models using gating is Long-Short Term Memory (LSTM)[23] network, defined by following equations (2-6):

$$r_t = \sigma_{sig}(W_r x_t + U_r h_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma_{sig}(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma_{sig}(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

$$c_t = r_t \circ c_{t-1} + i_t \circ \sigma_{tanh}(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

$$h_t = o_t \circ \sigma_{tanh}(c_t) \tag{6}$$

where r_t, i_t and o_t are the outputs of gates, i.e. standard log-sigmoid recurrent layers, each with two corresponding weight matrices (W_r, U_r, W_i, U_i, W_o, U_o) and a bias vector (b_r, b_i, b_o). c_t represents the cell memory state, with $\circ$ denoting element-wise multiplication, and is calculeted using another two weigth matrices W_c, U_c and a bias vector b_c.

A simplified gated model able to achieve results similar to LSTM was proposed in [24]. Gated Recurrent Unit (GRU) differs from LSTM in that it does not differentiate between cell memory state and output, thus reducing the internal complexity of a unit. GRU is defined by following equations (7-10):

$$z_t = \sigma_{sig}(W_z x_t + U_z h_{t-1} + b_z) \tag{7}$$

$$r_t = \sigma_{sig}(W_r x_t + U_r h_{t-1} + b_r) \tag{8}$$

$$c_t = r_t \circ h_{t-1} \tag{9}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_{tanh}(W_h x_t + U_h c_t + b_h) \tag{10}$$

Unlike LSTM, GRU does not use a separate layer for memory state, and its output is dependent only on the current input and previous output value. GRU uses two gates z_t and r_t. c_t represents the previous output after passing through a reset gate and does not have to be stored between timesteps.

GRU architecture reduces the number of required weight matrices to six (W_z, U_z, W_r, U_r, W_h, U_h) and bias vectors to three (b_r, b_i, b_o). An empirical comparison [25] of GRU and LSTM has shown that both perform similarly while outperforming standard recurrent neural networks. This makes GRU our architecture of choice, as it can achieve good performance while being relatively simpler compared to the LSTM.

IV. Semantic Embedding using Gated Recurrent Units

We consider a set of n training samples $X = (X_1, X_2, ..., X_n)$, where every training sample is a series of vectors (e.g., a spectrogram of the file or a series of Mel Frequency Cepstral Coefficients). Each training sample is annotated with a vector which we will denote as a_i for i-th training sample. The annotation vector consists of multiple real numbers for regression problems (e.g., continuous emotional scales), or $k-1$ zeroes and a single one for k-class classification (e.g., categorical models of emotion). We will use A to denote an annotation matrix, i.e., a matrix formed by stacking annotation vectors as rows.

Training sample $X_i = (x_{i1}, x_{i2}, ..., x_{il_i})$ is a series containing l_i vectors of the same size. Our goal is to find a representation of the training data that can represent a series X_i in a concise form of a vector f_i.

Let $H_i = (h_{i1}, h_{i2}, ...h_{il_i})$ be the output of a GRU resulting from input sequence X_i, according to equations (7-10). A feature vector f_i describing the sequence X_i is defined as a mean over the output sequence of the neural network (11):

$$f_i = \sum_{j=1}^{l_i} \frac{h_{ij}}{l_i} \tag{11}$$

Let F be a matrix containing n representations of music files and A be the annotation matrix. We assume ordering of rows in matrix F corresponds to the ordering of rows in matrix A, that is, i-th row of matrix F corresponds to the same music file as i-th row of matrix A.

We want to define a loss function that allows us to find a feature space in which similarity between vectors representing two music pieces corresponds to the similarity between their annotations. A simple way to define similarity between annotations is cosine similarity (12), commonly used in document retrieval:

$$cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \|y\|} \tag{12}$$

Cosine similarity is bound between 1 and -1 regardless of the vector dimensionality, which is a property we seek since annotation space can be of a different dimension than the output feature space. Let F' and A' be row-normalized matrices F and A respectively, i.e., if i-th row of the F matrix is f_i, then i-th row of the F' matrix equals to $\frac{f_i}{\|f_i\|}$. Loss function (13) can be then defined as:

$$L(F, A) = \|F' F'^{\top} - A' A'^{\top}\| \tag{13}$$

With a loss function defined this way, weights can be optimized using a gradient descent method.

Learned features are used as an input for a separate classifier, as shown on the right in Fig. 1. In contrast, a standard RNN depicted on the left uses an additional output layer y, defining an output (14) for i-th sequence as:

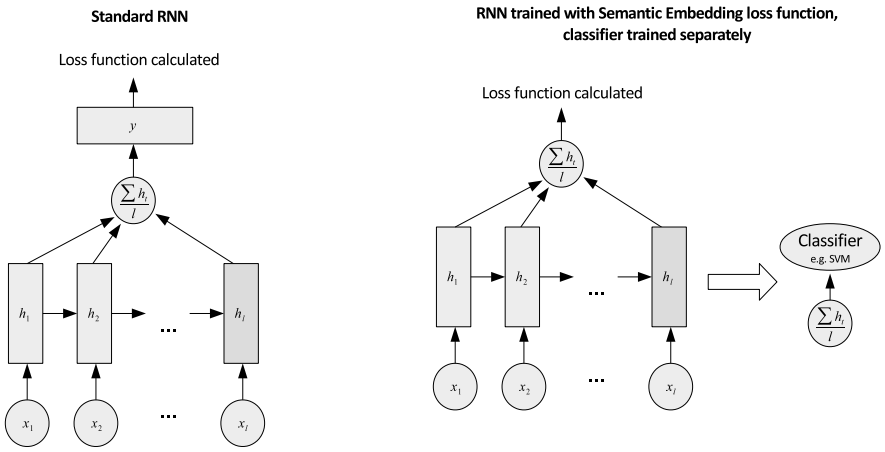

Fig. 1. Semantic embedding approach compared to standard RNN approach. Rectangles denote neural network layers. Output passes through one layer less before loss function calculation, making gradient calculation more robust.

$$y_i = \sigma(W_y f_i + b_y) \tag{14}$$

and then minimizing mean square error between y_i and a_i.

V. Experiments

The goal of our experiments is to compare the results of the proposed approach to both previously reported results achieved with standard features on emotion recognition datasets and a typical RNN which doesn't separate feature learning from the classifier. Since our feature learning approach should be general and usable in both classification and regression problems, we select two publicly available benchmark datasets, one for classification and one for regression.

The datasets are similar in that they were gathered through crowdsourcing, and represent the task of predicting a community consensus.

Another motivation for selecting the particular datasets is that both were employed in papers exploring the selection of features for music emotion recognition tasks [27][17]. Since our purpose is to establish whether our feature learning method can produce comparable or even better set of features than that based on domain knowledge, results reported by these papers provide a good point of reference. We report our results using the same figures of merit as the cited papers.

A. Datasets

The first dataset [26] on which we perform experiments uses the GEMS model of emotions. Magnatune record label supplied the songs which were selected to represent a variety of artists and musical styles. The annotations were gathered using a facebook game Emotify, in which respondents were asked to tag a music piece with at least 3 of the 9 mid-level GEMS emotions. All of the responses for each song are available. Our annotations take a form of 9-element vectors with real values in the range [0,1] indicating the percentage of respondents who tagged the music piece with particular emotion. The music files in the dataset are 1 minute long. There are 400 files in the dataset, split evenly between four music genres: rock, pop, electronic and classical.

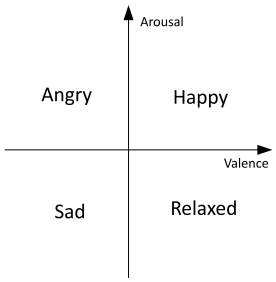

Fig. 2. Four classes in Lastfm dataset as VA plane quadrants

The second dataset, LastFM [27], contains music files split into four classes: angry, happy, sad and relaxed. The annotations were created based on 30 most popular Lastfm tags containing the name of emotion (e.g. "sad rock", "songs that make me happy"), with some unrelated tags excluded (e.g., tag "happysad" which corresponds to a name of a Polish band rather than emotional content). Music files for songs in the tags were originally obtained from 7digital.com website. It is important to note that classes in this dataset correspond to four quadrants of a Valence-Arousal plane (see Fig.2), which gives us additional information about the similarity between music pieces. Using the Semantic Embedding approach, we can capitalize on this information. We annotate files with two-dimensional vectors corresponding to the position on the V-A plane: (1,1) for happy, (1,-1) for relaxed, (-1,1) for angry and (-1,-1) for sad.

The music files are either 30 seconds or one minute long, and were not grouped by genres. There are 638 files in the "angry" class, 752 in "happy", 749 in "relaxed" and 763 in

"sad".

B. Conditions of the Experiments

As neither of the two datasets is split into training and test data, we use 10-fold cross-validation to measure the performance of proposed approach. Our implementation uses the Theano python library [31] which allows for efficient GPU usage and easy gradient computation. The representation of music files we use for the purposes of feature learning is a spectrogram with Mel frequency scale (40 bins) and log-scale for power. Spectrogram was extracted using default parameters of MIRToolbox MATLAB toolbox: frames are 50 ms long with 25 ms overlap. The input vector sequence of our RNN contains two frames of the spectrogram and deltas (changes from the previous frame) for these frames in each vector. This results in a sequence of 600 vectors of size 160 for a 30 seconds music file.

On a consumer-grade Nvidia GPU (GFX 970), training takes approximately 4.5 minutes for 100 epochs of optimization on 90% of the Emotify dataset. Computation time grows linearly with the number of files, resulting in approximately 40 minutes for training on 90% of the Lastfm dataset.

1) Classification and Regression: For classification and regression on learned features, we respectively use SVM and SVR [28] algorithms, implemented in sklearn python package [30]. For both of these algorithms, Radial Basis Function kernel (15) is used:

$$K(x, y) = e^{-\gamma \|x-y\|^2} \tag{15}$$

γ is a kernel parameter. The choice of methods is dictated both by their good performance and the need of comparison: previous research on both datasets was conducted using SVM for Lastfm dataset and SVR for Emotify dataset.

2) Parameter Selection: We set the size of our neural network to two recurrent GRU layers, 100 neurons in the first layer and 50 in the second. We use the adaptive learning rate method Adadelta [29]. The parameter ρ which governs the decay of previous updates' influence on the learning rate is set to 0.9. Initial weights are drawn from the Gaussian distribution with mean 0 and standard deviation 0.01. The network is trained on batches of 100 music files. For increased learning speed and as a form of preventing overfitting, in each epoch we only select a fragment of song of 100 vectors length (approximately 5 seconds) from each file, with a random selection of the starting point.

C. Experimental Results

The results of regression on the Emotify game dataset are shown in Table I. Following [16] we measure Pearson's correlation coefficient between predicted and ground truth values.

We compare standard GRU network and Semantic Embedding GRU network with SVR to results reported in existing literature. As a baseline we use the performance of best set of features found by [16] and our previous result achieved by a non-recurrent neural network with sparsity constraints [18].

TABLE II
ACCURACY OF EMOTION CLASSIFICATION ON THE LASTFM DATASET - GRU FEATURE LEARNING COMPARED TO RESULTS ACHIEVED WITH HIGH-LEVEL FEATURES REPORTED IN [27]

	Accuracy	**Vector size**
Dynamic Features + SVM [27]	0.372	7
Rhythm Features + SVM [27]	0.375	5
Harmonic Features + SVM [27]	0.475	10
Spectral Features + SVM [27]	0.519	32
Combined Best Features + SVM [27]	0.540	49
Standard GRU	0.501	50
Semantic Embedding GRU + SVM	**0.542**	50

As can be seen, a combination of Semantic Embedding GRU as a feature learner with SVR as a regressor visibly outperforms other approaches, while standard GRU trained with backpropagation to minimize mean square error cannot achieve comparable results. Our learned features are worse than features based on domain knowledge only in detection of solemnity and sadness. The latter may be related to the huge effect of harmony on the perceived sadness (notably, a common concept in European music is that minor chords are sad). This may be seen as an indication that our method cannot create features complex enough to express the concept of harmony.

The results of classification on Lastfm emotion classification dataset are shown in Table II. We compare the classification accuracy of the proposed feature learning approach and a standard GRU to results reported in [27]. The size of output feature vector is chosen to be comparable to the size of feature vectors based on domain knowledge.

As can be seen in Table II, our method achieves results slightly better that the best combination of hand-crafted features while not increasing the size of feature vector significantly.

VI. CONCLUSIONS AND FUTURE WORK

We presented a feature learning approach to the task of music emotion recognition inspired by the concept of Semantic Embedding present in document retrieval. We adapted the idea of learning a representation based on known similarities between sample data to use with recurrent neural network and multidimensional emotional annotations. Support vector machine was used on the learned features as a classification tool. The approach was tested on two datasets, both of which had been a subject of previous research concerning feature selection for music emotion recognition based on crowdsourced annotations.

The results confirm the usefulness of feature learning approach to music emotion recognition. Our method can create compact representations of music files using a GRU neural network. As can be seen in Table I and II, these representations allow us to achieve results better than those in existing literature after replacing standard music features with our learned features, while using the same classifier and regressor.

TABLE I
PREDICTIVE PERFORMANCE OF A RECURRENT NETWORK COMPARED TO PREVIOUS RESULTS ON EMOTIFY DATASET (PEARSON'S R)

Emotion Label	MIRtoolbox features + SVR [17]	Autoencoder + SVR [18]	Standard GRU	Semantic Embedding GRU + SVR
Amazement	0.16	**0.29**	0.28	**0.29**
Solemnity	0.43	0.50	0.47	**0.53**
Tenderness	**0.57**	0.54	0.50	0.53
Nostalgia	0.45	0.50	0.50	**0.54**
Calmness	0.50	**0.56**	0.51	**0.56**
Power	**0.56**	0.53	0.49	**0.56**
Joyful activation	**0.66**	0.53	0.63	**0.66**
Tension	0.46	0.48	0.46	**0.50**
Sadness	**0.42**	0.33	0.38	**0.42**
Average	0.47	0.47	0.47	**0.51**

The main limitation of our research so far is a low depth of the architecture and small timescales we have tested our approach on. The research can be expanded to test whether performance improvements can be achieved with additional layers and a higher number of neurons. Another area in which the research could be expanded is using convolutional layers inside the recurrent neural network.

REFERENCES

[1] C. C. Pratt, "Music as the language of emotion", *Bulletin of the American Musicological Society*, vol. 11, pp. 67-68, 1948.

[2] K. R. Scherer, M. Zentner, "Emotion effects of music: Production rules", *Music and emotion: Theory and research*, pp. 361-392, Oxford University Press, 2001.

[3] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbul, "Music emotion recognition: a state of the art review," *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 255-266, 2010.

[4] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," in *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pp. 403-408, 2012.

[5] N. Glazyrin , "Mid-level features for audio chord recognition using a deep neural network," *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*, vol. 155, no. 4, pp. 109-117, 2013.

[6] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6959-6963, 2014.

[7] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 565-570, 2012.

[8] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *(ISMIR)*, vol. 11, no. 445, 2011.

[9] Y. Vaizman, B. McFee, and G. Lanckriet, "Codebook based audio feature representation for music information retrieval," *IEEE/ACM Transactions on Acoustics, Speech and Signal Processing*, vol. 22, no. 10, pp. 1483-1493, 2014.

[10] P. Ekman, "An argument for basic emotions," *Cognition Emotion* vol. 6, no. 3, pp. 169-200, 2001.

[11] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbul, "Music emotion recognition: a state of the art review," *Proceedings of 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 255-266, 2010.

[12] J. Skowronek, M. McKinney, and S. van de Par, "A demonstrator for automatic music mood estimation," in *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, pp. 345-346, 2007.

[13] C. Laurier, O. Lartillot, T. Eerola, and P. Toiviainen: "Exploring Relationships between Audio Features and Emotion in Music," *Conference of European Society for the Cognitive Sciences of Music (ESCOM)*, 2009.

[14] U. Schimmack and R. Reisenzein, "Experiencing activation: energetic arousal and tense arousal are not mixtures of valence and activation," *Emotion*, vol. 2, no. 4, p. 412, 2002.

[15] Y. H. Yang, Y. C. Lin, Y. F. Su, and H. H. Chen, "A Regression Approach to Music Emotion Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 2, pp. 448-457, 2008.

[16] M. Zentner, D. Grandjean, and K. R. Scherer: "Emotions evoked by the sound of music: characterization, classification, and measurement," *Emotion*, vol. 8, no. 4, pp. 494-521, 2008.

[17] A. Aljanaki, F. Wiering, and R. Veltkamp, "Computational modeling of induced emotion using GEMS," *Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 373-378, 2014.

[18] J. Jakubik, H. Kwasnicka, "Sparse Coding Methods for Music Induced Emotion Recognition", *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 53-60, 2016.

[19] C. Wang, S. Yan, L. Zhang, H.-J. Zhang, Multi-label Sparse Coding for Automatic Image Annotation, *Proceedings of IEEE Intenational Conference on Computer Vision and Pattern Recognition*, pp. 1643-1650, 2009.

[20] Hao Wu, Martin Renqiang Min, and Bing Bai, " Deep semantic embedding", *Proceedings of SMIR@SIGIR*, pp. 4652, 2014.

[21] Jason Weston, Frdric Ratle, Hossein Mobahi, Ronan Collobert, "Deep Learning via Semi-Supervised Embedding", *Lecture Notes in Computer Science*, vol 7700, pp. 639-655, 2012.

[22] Goller, C.; Kuchler, A. "Learning task-dependent distributed representations by backpropagation through structure". *IEEE International Conference on Neural Networks*, vol. 1, pp. 347-352, 1996.

[23] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol 9, no. 8, pp. 1735-1780, 1997.

[24] D. Bahdanau, K. Cho, Y. Bengio "Neural Machine Translation by Jointly Learning to Align and Translate", *International Conference on Learning Representations*, 2015.

[25] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *arXiv:1412.3555*, 2014.

[26] A. Aljanaki, F. Wiering, R. C. Veltkamp. "Studying emotion induced by music through a crowdsourcing game", *Information Processing and Management*, vol. 52, no. 1, pp. 115-128, 2015.

[27] Y. Song, S. Dixon, M. Pearce "Evaluation of Musical Features for Music Emotion Classification", *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 523-528, 2012.

[28] A. J. Smola, B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing,* vol. 14, no. 3 , pp 199-222, 2004.

[29] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method", *arXiv:1212.5701*, 2012.

[30] F. Pedregosa et. al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[31] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions", *arXiv:1605.02688*, 2016.