Regression-based Music Emotion Prediction using Triplet Neural Networks

Kin Wai Cheuk, Yin-Jyun Luo
Information Systems,
Technology, and Design
Singapore University
of Technology and Design

Institute of High Performance
*Computing, A*STAR*
Singapore
kinwai_cheuk@mymail.sutd.edu.sg
yinjyun_luo@mymail.sutd.edu.sg

Balamurali B, T
Information Systems,
Technology, and Design
Singapore University
of Technology and Design
Singapore
balamurali_bt@sutd.edu.sg

Gemma Roig
Computer Science Department
Goethe University
Frankfurt am Main
Germany
roig@cs.uni-frankfurt.de

Dorien Herremans
Information Systems,
Technology, and Design
Singapore University
of Technology and Design

Institute of High Performance
*Computing, A*STAR*
Singapore
dorien_herremans@sutd.edu.sg

***Abstract*—In this paper, we adapt triplet neural networks (TNNs) to a regression task, music emotion prediction. Since TNNs were initially introduced for classification, and not for regression, we propose a mechanism that allows them to provide meaningful low dimensional representations for regression tasks. We then use these new representations as the input for regression algorithms such as support vector machines and gradient boosting machines. To demonstrate the TNNs' effectiveness at creating meaningful representations, we compare them to different dimensionality reduction methods on music emotion prediction, i.e., predicting valence and arousal values from musical audio signals. Our results on the DEAM dataset show that by using TNNs we achieve 90% feature dimensionality reduction with a 9% improvement in valence prediction and 4% improvement in arousal prediction with respect to our baseline models (without TNN). Our TNN method outperforms other dimensionality reduction methods such as principal component analysis (PCA) and autoencoders (AE). This shows that, in addition to providing a compact latent space representation of audio features, the proposed approach achieves higher performance than the baseline models.**

***Index Terms*—Triplet neural network (TNN), Music emotion recognition (MER), Support vector machine (SVM), Gradient boosting machine (GBM) , Dimensionality reduction, Regression**

I. Introduction

The link between music and emotions has been investigated extensively by cognitive scientists and musicologists over the years [1, 2, 3, 4], which has caused the emergence of the field of automatic music emotion recognition (MER). Being able to predict the emotion from a music audio clip has a myriad of applications, such as managing personal music collections [5], mood-based music recommendation [6, 7, 8, 9], musicology [10], and music therapy [11, 12]. In order to label emotions, researchers often use a two-dimensional arousal-valence (A/V) representation [13, 14]. Given that A/V values are continuous values, we will approach the emotion prediction task as a regression problem.

One of the challenges when tackling automatic emotion prediction from audio is to identify the ideal audio features that best capture the emotion evoked by the audio signal [15]. Aljanaki et al. [16] investigated the performance of 44 audio features extracted using MIRToolbox, PsySound, and SonicAnnotator. Out of these 44 features, they found that 26 features were ideal when predicting arousal and 27 features for valence. In a paper published by Weninger et al. [17], 6,373 features were to train a support vector machine model to predict A/V. Although the number of features used in this study is many times larger than the 44 [16], the improvement in R^2 score for estimating arousal values is marginal (0.65 versus 0.64). The R^2 score for valence, however, was found to be relatively better (0.42 versus 0.36) [17]. The main challenge when using such a large feature space is that it requires a lot of computational resources, or even faces scalability issues, as reported by Markov et al. [18]. When their model was trained with a lot less data, the performance of the regression was affected severely. An efficient dimensional reduction technique may provide a solution for this issue. In this paper, we tackle the MER problem by adopting a regression version of the triplet neural network structure in order to reduce the dimensionality, while at the same time creating a more meaningful representation, thus improving the result of the prediction compared to other dimensionality reduction techniques. The source code is available online[1].

II. Regression Using a TNN

A. Triplet neural networks

TNN is a neural network technique first proposed by researchers from Google in 2015 [19]. Since then it has been used for a variety of tasks such as face recognition or person identification [20, 21, 22, 23, 24, 25]. Originally, TNNs were used in classification problems to learn a new feature

[1] https://github.com/KinWaiCheuk/IJCNN2020_music_emotion

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

representation of data, so that this new representation can easily be disentangled with regard to the classes. TNNs consist of 3 inputs, namely anchor (A) input, positive (P) input, and negative (N) input. The positive input belongs to the same class as that of the anchor input while the negative input is from a different class than the anchor input. This triplet data is fed to a neural network which shares the same weights for each of these 3 inputs. The output of the TNN is then passed to a triplet loss function $L(A,P,N)$ as shown below.

$$L(A,P,N) = \max(D(A,P) - D(A,N) + \alpha, 0), \quad (1)$$

where $D(A,P)$ represents the squared Euclidean distance between the anchor vector v'_A and the positive example vector v'_P; $D(A,N)$ represents the squared Euclidean distance between v'_A and the negative example vector v'_N as defined in Figure 1; and α is the margin parameter that specifies how far positive and negative examples should be apart. The training goal is to reduce the distance between similar samples $D(A,P)$, and increase the distance for samples in different classes $D(A,N)$. In other words, we want to learn a representation that positions vectors that belong to the same class close together and those from a different class far apart. Note that the max operation in Eq. (1) is equivalent to the ReLu operator.

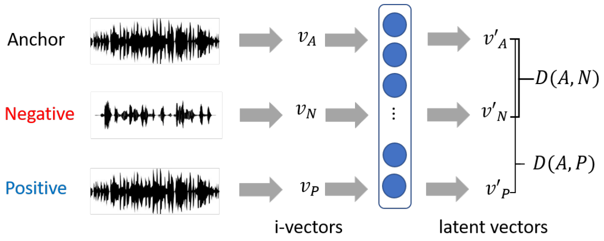

Fig. 1. Schematic diagram of the triplet neural network.

B. Defining positive and negative samples for regression

When dealing with a classification task, the definition of positive (or negative) samples is straightforward, i.e., those belonging (or not belonging) to the same class. In the case of regression, however, a new strategy is needed because it operates in the continuous space. Instead of a finite number of discrete classes, the labels now have an infinite number of possible values even in the range $[-1,1]$. Despite attempts in applying TNNs [26, 27] to regression datasets, it only works if the dataset is specially designed for this. In other words, their methods cannot be generalized to other datasets. For example, in Yang et al. [27]'s research, the positive and negative samples are defined by using two classes (pre-triage: before seeing the doctor; and post-triage: after seeing the doctor) of videos without using the true annotation (blood pressure) of the dataset. Without the explicit information about pre-triage and post-triage, the method would not work. Lu et al. [26]'s dataset already includes a label for the positive and negative samples, so they can directly form the triplets without any triplet mining. Although they successfully apply TNN to a regression problem, their method only works for this specific (labeled) dataset, and cannot be applied in our problem. For instance, if we have an anchor with value 0.1, should the sample with value 0.3 be a positive sample, or a negative sample? Setting absolute, discrete bins does not work well for regression. For example, let us define the interval $[0,0.5)$ to be bin A, and $[0.5,1]$ to be bin B. For the anchor sample $y_{v_a} = 0.5$, we need to decide if the two samples $y_{v_1} = 0.4$ and $y_{v_2} = 1$, are positive or negative samples. Under the discrete absolute binning scheme, y_{v_1} will be a negative sample of the anchor since they belong to different bins, while y_{v_2} will be a positive sample of the anchor since they are in the same bin. This could skew our results, as y_{v_1} is obviously closer to the anchor than y_{v_2}. Therefore, we propose a more effective method to define positive and negative samples in regression below. In this section, we describe our mechanism for defining positive and negative samples without explicit information on positive and negative examples.

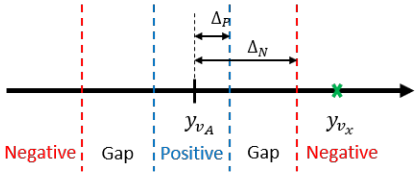

Fig. 2. Defining anchor, positive and negative samples. If the sample lies in the gap region, it will discarded and it will not be used for training.

In our sampling approach, instead of using a fixed binning, we define fixed threshold gaps that are applied on the anchor samples. Let the valence value for the anchor vector be y_{v_A}. Let Δ_P be the threshold value, such that for a vector with valence value $y_{v_X} \in [y_{v_A} - \Delta_P, y_{v_A} + \Delta_P]$, the sample is considered to be a positive example. Similarly, we define another threshold value Δ_N, such that the vector y_{v_X} is considered to be a negative sample if $y_{v_X} \in (-\infty, y_{v_A} - \Delta_N] \vee [y_{v_A} + \Delta_N, \infty)$. The difference between Δ_N and Δ_P form a 'gap' that guides the network to learn the more distinguishing samples first. The same process is used to define positive and negative arousal values. A schematic representation of the positive and negative sample definition is shown in Figure 2. In our experiments, we normalize arousal and valence values so that they fall within $[-1,1]$. The threshold values should be adjusted according to the data distribution. In our experiments, we set $\Delta_P = 0.1$ and $\Delta_N = 0.5$, which has been found to perform best through cross-validation.

III. Dimensionality reduction for emotion prediction

Most research on static emotion annotation (i.e., one rating per song) is based on the 2013 MediaEval dataset[2] (a subset of the DEAM dataset)[16, 17, 18, 28, 29, 30]. We therefore compare our results with prominent papers using this dataset, and we will call it MediaEval experiment. Then we also use

[2] http://cvml.unige.ch/databases/emoMusic/

the complete DEAM dataset to test our model, and we will call it DEAM experiment. We implement our novel TNN-regression approach for dimentionality reduction and combine it with both a support vector regressor (SVR) and gradient boosting machine (GBM) to solve the regression problem for the valence and arousal values jointly.

A. MediaEval experiment

For this experiment we use the 2013 MediaEval dataset[2]. It contains a total of 744 audio files with 6,669 features (listed in TABLE I.). Each audio file is labeled with A/V values [31]. Different statistical features such as maximum, minimum, mean, range, and etc. are included in the dataset for each of the base features in the dataset, resulting in 6,669 features in total.

TABLE I

List of openSMILE features provided by the MediaEval 2013 dataset. Different summary statistic features are provided for these base features, resulting in a total of 6,669 features [32]

Feature Name	Size
F0	117
F0env	117
mfcc	1,521
pcm_LOGenergy	117
pcm_Mag_fband0-250	117
pcm_Mag_fband0-650	117
pcm_Mag_fband1000-4000	117
pcm_Mag_fband250-650	117
pcm_Mag_fband3010-9123	117
pcm_Mag_melspec	3,042
pcm_Mag_spectralCentroid	117
pcm_Mag_spectralFlux	117
pcm_Mag_spectralMaxPos	117
pcm_Mag_spectralMinPos	117
pcm_Mag_spectralRollOff25.0	117
pcm_Mag_spectralRollOff50.0	117
pcm_Mag_spectralRollOff75.0	117
pcm_Mag_spectralRollOff90.0	117
pcm_zcr	117
voiceProb	117

The TNN implemented in this experiment consists of a single fully connected layer with 600 neurons and ReLU as the activation function. When using less neurons than 600, the model performance decays. With more neurons, there are too many features such that the classifiers cannot be trained effectively. The network was first trained using 50,000 triplet pairs sampled from the dataset. The mining of the 50,000 triplet pairs consists of the following steps:

1) Randomly pick a sample as the anchor v_A from the dataset, which has a label value of y_{v_A}.
2) Find the positive sample v_P for the anchor using the method mentioned in II-B.
3) Find the negative sample v_N for the anchor using the method mentioned in II-B.
4) A triplet is formed with the result from the previous steps (v_A, v_P, v_N).

TABLE II

List of openSMILE features when using the configuration file **IS13_ComParE_lld-func.conf**. Different summary statistic features are provided for these base features, resulting in a total of 260 features [29, 33].

Feature Name	Size
F0final	4
audSpec_Rfilt	104
audspecRasta_lengthL1norm	4
audspec_lengthL1norm	4
jitterDDP	4
jitterLocal	4
logHNR	4
pcm_RMSenergy	4
pcm_fftMag_fband1000-4000	4
pcm_fftMag_fband250-650	4
pcm_fftMag_mfcc	56
pcm_fftMag_psySharpness	4
pcm_fftMag_spectralCentroid	4
pcm_fftMag_spectralEntropy	4
pcm_fftMag_spectralFlux	4
pcm_fftMag_spectralHarmonicity	4
pcm_fftMag_spectralKurtosis	4
pcm_fftMag_spectralRollOff25.0	4
pcm_fftMag_spectralRollOff50.0	4
pcm_fftMag_spectralRollOff75.0	4
pcm_fftMag_spectralRollOff90.0	4
pcm_fftMag_spectralSkewness	4
pcm_fftMag_spectralSlope	4
pcm_fftMag_spectralVariance	4
pcm_zcr	4
shimmerLocal	4

5) Repeat steps 1 to 4 by choosing other samples as the anchor until the number of mined triplets has reached $50,000$.

With this mining method, we can obtain 50,000 triplet pairs from the original 744 individual audio files.

After an initial 10 epochs, another 50,000 triplet samples were generated and the process continued. In this way, we prevent the model from overfitting to a small set of triplet samples. In total, this was repeated 25 times (i.e., 250 epochs in total) until the model converges, thereby allowing the network to train on as many samples as possible. An Adam optimizer with learning rate 10^{-5} was used to minimize the triplet loss during training.

We compared our proposed method with other models applying to the same dataset. Markov et al. [18] trained a support vector regressor (SVR) using the features provided in the dataset. Their R^2 score, however, are only 0.112 and 0.300 for valence and arousal respectively (see Table III) [18]. They further reported scaling issues with their model due to the large number of audio features. [28] modified the GPR model to obtain a greater accuracy. Our approach also addresses the scalability issue reported by Markov et al. [18], by using a TNN to reduce the number of features from 6,669 features to 600. Both the original features (baseline model) and the transformed low dimensional vectors were used to train an SVR and GBM (referred to as TNN-SVR and TNN-GBM, respectively).

To evaluate the ability of learning an effective latent space, we compared the TNN results against other dimensionality reduction techniques such as principal component analysis (PCA), Gaussian random projection (RP), and a neural network-based autoencoder (AE). For PCA and RP, the number of components in the transformed space was set to 600 so as to match the number of neurons of the TNN; and the random state for the RP was set to 50. For AE, 600 neurons with ReLU activation were used (because our TNN model also uses 600 neurons) for the encoder, and the auto-encoder was trained for until convergence (around 100 epochs). We used standard R^2 scores with ten fold cross-validation. For each fold, the weights of the TNN were reset and retrained.

B. DEAM experiment

The setup for this experiment is similar to MediaEval experiment, except that we use a different dataset with different TNN configuration. The state-of-the-art comparison models remain the same. The DEAM dataset contains 1,724 songs [29, 33] labeled with A/V values. The dataset provides a total of 260 features extracted from the audio clips using the configuration file **IS13_ComParE_lld-func.conf**. The list of features extracted with the configuration file is listed in TABLE II. For each of the feature, different statistics are calculated such as means, standard deviations, first derivatives, and different bands (for MFCC and specRfilt), resulting in 260 openSMILE features in total. The objective of this experiment is to show the TNN's ability of perform efficient dimensionality reduction on a larger dataset.

We focus on learning a latent space with TNNs and use the learned latent space as the model input for the SVM and GBM regressor. Two different sets of TNN parameters were tested in this experiment: 1) 100 neurons (around one half of the original features); and 2) 50 neurons (around one fourth of the original features) in the fully connected layer. The training procedure is same as in MediaEval experiment except that 150,000 samples were generated each time. Other parameters are kept the same as in MediaEval experiment.

IV. Discussion

A. Visualization of learned embedded spaces.

A visualisation of the data distribution before and after the TNN, PCA, and AE transformation is presented in Figure 3. A t-distributed stochastic neighbor embedding (t-SNE) was used to project the data into two-dimensions [34]. We divided the songs into 4 classes for visualization, namely, high arousal, mid-high arousal, mid-low arousal, and low arousal values. The 100 songs with the highest arousal values were selected as the high arousal class, and the 100 songs with lowest arousal values as the low arousal class. We then split the remaining songs into mid-high and mid-low arousal classes using a similar procedure. The top left corner of Figure 3 shows the data distribution when projecting 6,669 features into a two-dimensional plane by using t-SNE. The overlap between classes is considerable. The top right corner of Figure 3 corresponds to the t-SNE projection on the TNN learned latent space for the data. It has more obvious clusters with only minor overlap. The bottom of the figure shows the data distribution when using PCA and AE for feature reduction. No clear clusters are formed under these two transformations.

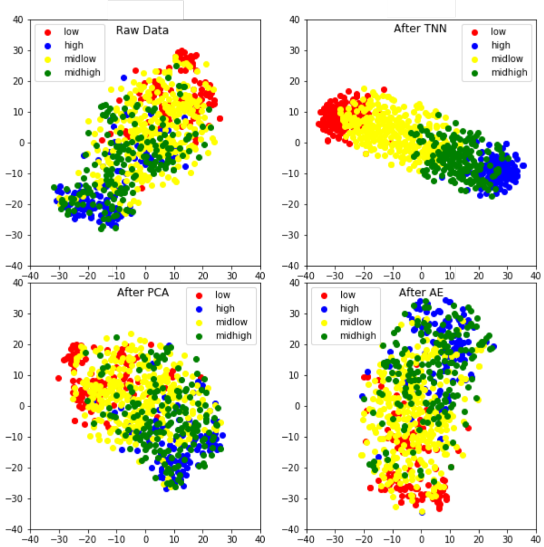

Fig. 3. t-SNE visualization of the 6,669 MediaEval features and the 600 features after reduction by TNN, PCA, and AE.

B. Result for MediaEval experiment

In our experiments, we study if we can still maintain a relatively good regression result, with reduced features learned by the TNN. From our experiment, we see that TNN-based models performed best when less layers were used. Therefore, we used a single layer fully connected network with ReLU activation as our TNN structure. When comparing the results of our TNN method with other dimension reduction techniques (see Table III), We can see that the TNN has the ability to significantly reduce dimensionality of the data (more than 10%, to 600 features), while still maintaining a relatively high R^2 score for both valence and arousal compared with the model using original features. Traditional methods such as PCA and RP were not able to deliver a similar performance with the reduced features. Although the performance of AE is marginally better than PCA, the resulting R^2 scores cannot match the performance of the TNN-SVR and TNN-GBM model. In order words, our proposed TNN method is more suited for reducing the dimension of the data in preparation of regression, when compared to traditional methods such as PCA, RP, and AE. We should note that TNN is a supervised clustering method as opposed to unclustered methods such as PCA, RP, and AE. Since we are assessing the potential of different dimension reduction techniques as a precursor to perform a supervised classification problem, we argue that using the labels to perform the dimentionality reduction is a valid approach, as those labels are used for learning the final prediction.

The proposed TNN-SVR and TNN-GBM outperform Markov et al. [18]'s model for both arousal and valence. Only in the case of valence, does the GPR [28] reach a higher R^2 than our proposed model. It is important to note here that the number of features was reduced from 6,669 by one fold to 600, while still maintaining a relatively high R^2 score for both valence and arousal values. Readers should note that Markov et al. [18] used seven fold cross-validation in their study instead of ten fold cross-validation. Since, the exact composition of the folds used for evaluation are unknown, it would therefore be worth exploring in future research if a TNN combined with exact set up as [18] and [28] would achieve further improvements in performance.

TABLE III
RESULTS FOR MEDIAEVAL EXPERIMENT: R^2 SCORES ± STANDARD DEVIATION FOR VARIOUS MODELS ON THE MEDIAEVAL 2013 DATASET.

	Valence	Arousal
SVR[18]	0.112	0.300
GPR[18]	0.170	0.581
GPR [28]	0.413 ± 0.043	0.636 ± 0.040
GBM(original features)	0.431 ± 0.089	0.662 ± 0.057
PCA-GBM (600 features)	0.251 ± 0.118	0.566 ± 0.063
RP-GBM (600 features)	0.229 ± 0.152	0.619 ± 0.076
AE-GBM (600 features)	0.236 ± 0.145	0.578 ± 0.077
TNN-GBM (600 features)	**0.374 ± 0.058**	**0.621 ± 0.080**
SVR (original features)	0.347 ± 0.086	0.614 ± 0.054
PCA-SVR (600 features)	0.087 ± 0.119	0.224 ± 0.140
RP-SVR (600 features)	0.334 ± 0.097	0.608 ± 0.062
AE-SVR (600 features)	0.280 ± 0.150	0.598 ± 0.100
TNN-SVR (600 features)	**0.378 ± 0.066**	**0.638 ± 0.055**

TABLE IV
RESULTS FOR DEAM EXPERIMENT: R^2 SCORES ± STANDARD DEVIATION FOR VARIOUS MODELS ON THE DEAM DATASET.

	Valence	Arousal
SVR (260 features)	0.324 ± 0.133	0.638 ± 0.062
GBM (260 features)	0.318 ± 0.151	0.678 ± 0.062
PCA-GBM (100 features)	0.288 ± 0.123	0.584 ± 0.091
RP-GBM (100 features)	0.251 ± 0.137	0.573 ± 0.079
AE-GBM (100 features)	0.312 ± 0.109	0.623 ± 0.067
TNN-GBM (100 features)	**0.367 ± 0.113**	**0.662 ± 0.065**
PCA-GBM (50 features)	0.251 ± 0.124	0.576 ± 0.091
RP-GBM (50 features)	0.215 ± 0.149	0.549 ± 0.086
AE-GBM (50 features)	0.255 ± 0.146	0.601 ± 0.076
TNN-GBM (50 features)	**0.339 ± 0.124**	**0.661 ± 0.068**
PCA-SVR (100 features)	0.274 ± 0.121	0.550 ± 0.058
RP-SVR (100 features)	0.270 ± 0.125	0.615 ± 0.061
AE-SVR (100 features)	0.311 ± 0.119	0.629 ± 0.070
TNN-SVR (100 features)	**0.361 ± 0.112**	**0.672 ± 0.065**
PCA-SVR (50 features)	0.195 ± 0.119	0.444 ± 0.063
RP-SVR (50 features)	0.203 ± 0.017	0.574 ± 0.077
AE-SVR (50 features)	0.269 ± 0.159	0.606 ± 0.065
TNN-SVR (50 features)	**0.352 ± 0.112**	**0.669 ± 0.070**

C. Result for DEAM experiment

The experiments also show that GBM is more effective when dealing with the dataset that has a larger feature space (the MediaEval dataset), since it has a much higher R^2 score for both valence and arousal, compared to the GBM built on a reduced feature set. On the dataset with less features (the DEAM dataset), the TNN dimension reduction marginally improves the GBM performance (see Table IV). SVR, on the other hand, is much more effective when dealing with features after dimension reduction. When using SVR, the TNN can achieve a 90% feature reduction with a 9% improvement in valence prediction R^2 and 4% improvement in arousal prediction R^2. When using GBM, the TNN can achieve the same feature reduction with 13% decrease in valence prediction R^2 and 6% decrease in arousal prediction R^2. For both regression algorithms, the TNN outperforms all other traditional dimensionality reduction algorithms such as PCA, RP, and AE.

We show that TNNs can further reduce the 260 DEAM features to only 100 or 50 features without a huge impact on the regression accuracy. In the case of SVR with reduced TNN features, the R^2 scores for both valence and arousal still outperform the baseline SVR model. Among PCA, RP, and AE, PCA performs the worst, probably because it is a linear process, while the other 2 can capture non-linearity.. When the number of features is reduced from 260 to 50, PCA-SVR has a 40% decrease in R^2 scores for valence, and a 30% decrease for arousal. Although AE with either SVR or GBM performs better than PCA, it still has a 17% and 5% decrease in valence and arousal R^2 scores when the dimension is reduced to 50. TNNs, on the other hand, are able to better capture the non-linear relations among the original features, thus, successfully reducing the features while improving the valence and arousal prediction R^2 score by 8% and 5% respectively. A similar trend can be observed for the GBM-TNN case, despite a slight 2.5% decrease in R^2 score for arousal value when the TNN's layer size is reduced to 50.

V. CONCLUSION

We propose a strategy to leverage triplet neural networks for regression tasks with a new adaptative mining of negative and positive samples, And we show its efficiency on music emotion regression. Based on two experiments (on MediaEval and DEAM datasets), we see that our hybrid TNN, combined with a SVR or GBM regressor, has the ability to perform significant dimensionality reduction while still improving the R^2 scores. Traditional methods such as principal component analysis, and autoencoders failed to maintain the same level of accuracy. We believe that this paper provides a foundation for a deeper and more comprehensive future study of applying TNNs to regression problems. Some of the open questions that we aim at further investigating are: (1) Can our TNN reduce the latent space to only 3 or 2 components? Why do spectrogram features result in a less effective TNN latent space than the openSMILE features?

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their constructive reviews. This work is supported by a Singapore International Graduate Award (SINGA) provided by the Agency for Science, Technology and Research (A*STAR), under grant no. SING-2018-02-0204. Moreover, the equipment required to run the experiments is supported by SUTD-MIT IDC grant no. IDG31800103 and MOE Grant no. MOE2018-T2-2-161.

References

[1] C. C. Pratt, "Music as the language of emotion." The Library of Congress, 1952.

[2] P. N. Juslin and J. A. Sloboda, *Music and emotion: Theory and research.* Oxford University Press, 2001.

[3] D. Herremans and E. Chew, "Tension ribbons: Quantifying and visualising tonal tension." in *Proc. of the int. conf. on technologies for music notation and representation*, Cambridge, UK, 2016.

[4] H. T. P. Thao, D. Herremans, and G. Roig, "Multimodal deep models for predicting affective responses evoked by movies," 2019.

[5] S. J. Cunningham, M. Jones, and S. Jones, "Organizing digital music for use: an examination of personal music collections." ISMIR, 2004.

[6] B.-J. Han, S. Rho, S. Jun, and E. Hwang, "Music emotion classification and context-based music recommendation," *Multimedia Tools and Applications*, vol. 47, no. 3, pp. 433–460, 2010.

[7] F.-F. Kuo, M.-F. Chiang, M.-K. Shan, and S.-Y. Lee, "Emotion-based music recommendation by association discovery from film music," in *Proc. of the 13th annual ACM int. conf. on Multimedia.* ACM, 2005, pp. 507–510.

[8] C. Inskip, A. Macfarlane, and P. Rafferty, "Towards the disintermediation of creative music search: analysing queries to determine important facets," *International Journal on Digital Libraries*, vol. 12, no. 2-3, pp. 137–147, 2012.

[9] M. I. Mandel, G. E. Poliner, and D. P. Ellis, "Support vector machine active learning for music retrieval," *Multimedia systems*, vol. 12, no. 1, pp. 3–13, 2006.

[10] D. Herremans, S. Yang, C.-H. Chuan, M. Barthet, and E. Chew, "Imma-emo: A multimodal interface for visualising score-and audio-synchronised emotion annotations," in *Proc. of the 12th Int. Audio Mostly Conf.* ACM, 2017, p. 11.

[11] O. Sourina, Y. Liu, and M. K. Nguyen, "Real-time eeg-based emotion recognition for music therapy," *Journal on Multimodal User Interfaces*, vol. 5, no. 1-2, pp. 27–35, 2012.

[12] A. Huq, J. P. Bello, A. Sarroff, J. Berger, and R. Rowe, "Sourcetone: An automated music emotion recognition system," in *Proc. of the int. conf. on Music Information Retrieval*, 2009.

[13] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[14] R. E. Thayer, *The biopsychology of mood and arousal.* Oxford University Press, 1990.

[15] A. Huq, J. P. Bello, and R. Rowe, "Automated music emotion recognition: A systematic evaluation," *Journal of New Music Research*, vol. 39, no. 3, pp. 227–244, 2010.

[16] A. Aljanaki, F. Wiering, and R. C. Veltkamp, "Mirutrecht participation in mediaeval 2013: Emotion in music task." in *MediaEval.* Citeseer, 2013.

[17] F. Weninger, F. Eyben, and B. Schuller, "The tum approach to the mediaeval music emotion task using generic affective audio features," in *Proceedings MediaEval 2013 Workshop, Barcelona, Spain*, 2013.

[18] K. Markov, M. Iwata, and T. Matsui, "Music emotion recognition using gaussian processes." in *MediaEval*, 2013.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[20] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2353–2367, 2016.

[21] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 1335–1344.

[22] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *European conf. on computer vision.* Springer, 2016, pp. 475–491.

[23] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 1288–1296.

[24] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognition*, vol. 48, no. 10, pp. 2993–3003, 2015.

[25] K. W. Cheuk, G. Roig, D. Herremans *et al.*, "Latent space representation for multi-target speaker detection and identification with a sparse dataset using triplet neural networks," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.

[26] R. Lu, K. Wu, Z. Duan, and C. Zhang, "Deep ranking: Triplet matchnet for music metric learning," in *2017 IEEE int. conf. on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2017, pp. 121–125.

[27] H.-C. Yang, F.-S. Tsai, Y.-M. Weng, C.-J. Ng, and C.-C. Lee, "A triplet-loss embedded deep regressor network for estimating blood pressure changes using prosodic features," in *2018 IEEE int. conf. on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 6019–

6023.

[28] S. Fukuyama and M. Goto, “Music emotion recognition with adaptive aggregation of gaussian process regressors,” in *int. conf. on Speeh and Signal Processing (ICASSP)*. IEEE, 2016, pp. 71–75.

[29] M. Soleymani, M. N. Caro, E. M. Schmidt, C.-Y. Sha, and Y.-H. Yang, “1000 songs for emotional analysis of music,” in *Proc. of the 2Nd ACM International Workshop on Crowdsourcing for Multimedia*, ser. CrowdMM 13. New York, NY, USA: ACM, 2013, pp. 1–6.

[30] F. Eyben, F. Weninger, F. Gross, and B. Schuller, “Recent developments in opensmile, the munich open-source multimedia feature extractor,” in *Proc. of the 21st ACM int. conf. on Multimedia*, ser. MM 13. New York, NY, USA: ACM, 2013, pp. 835–838.

[31] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi *et al.*, “The interspeech computational paralinguistics challenge,” in *Proc. INTERSPEECH, Lyon, France*, 2013.

[32] M. Soleymani, M. N. Caro, E. M. Schmidt, and Y.-H. Yang, “The mediaeval 2013 brave new task: Emotion in music.” in *MediaEval*. Citeseer, 2013.

[33] A. Alajanki, Y.-H. Yang, and M. Soleymani, “Benchmarking music emotion recognition systems,” *PLOS ONE*, pp. 835–838, 2016.

[34] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.