INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

EigenEmo: Spectral Utterance Representation Using Dynamic Mode Decomposition for Speech Emotion Classification

Shuiyang Mao, P. C. Ching, Tan Lee

Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong
maoshuiyang@link.cuhk.edu.hk, pcching@ee.cuhk.edu.hk, tanlee@ee.cuhk.edu.hk

Abstract

Human emotional speech is, by its very nature, a variant signal. This results in dynamics intrinsic to automatic emotion classification based on speech. In this work, we explore a spectral decomposition method stemming from fluid-dynamics, known as Dynamic Mode Decomposition (DMD), to computationally represent and analyze the global utterance-level dynamics of emotional speech. Specifically, segment-level emotion-specific representations are first learned through an Emotion Distillation process. This forms a multi-dimensional signal of emotion flow for each utterance, called Emotion Profiles (EPs). The DMD algorithm is then applied to the resultant EPs to capture the eigenfrequencies, and hence the fundamental transition dynamics of the emotion flow. Evaluation experiments using the proposed approach, which we call EigenEmo, show promising results. Moreover, due to the positive combination of their complementary properties, concatenating the utterance representations generated by EigenEmo with simple EPs averaging yields noticeable gains.

Index Terms: speech emotion classification, dynamic mode decomposition, emotion distillation, emotion profile

1. Introduction

Human emotional speech is dynamic, which, when taken into account, may augment our understanding of these complex signals and lead to modeling advancements. In the past few decades, efforts have been made to perform dynamic modeling for automatic speech emotion classification explicitly. Dynamic models, such as hidden Markov models (HMMs) [1, 2, 3] and recurrent neural networks (RNNs), e. g., with long short-memory (LSTM) [4, 5, 6, 7, 8], are frequently used. For their input features, the common practice considers frame-based low-level features such as Mel Frequency Cepstrum Coefficients (MFCCs), energy, or pitch. The final assignment of an emotion label is then based on the *low-level feature fluctuations* captured by the dynamic models.

As a complement to most of the work as mentioned above, this work aims at utilizing spectral methods for the dynamic modeling of emotion. Spectral analysis is widely used in signal processing to decompose a signal into its component frequencies, thereby revealing the dominant dynamics that make up the signal and summarizing its transitions. In particular, this paper presents the Dynamic Mode Decomposition (DMD) [9, 10, 11] algorithm to identify the dominant behavior that underlies emotional speech. The DMD algorithm was invented by P. Schmid as a diagnostic tool for extracting dynamic information from temporal measurements of a multivariate fluid flow vector. The dynamic modes extracted are the non-orthogonal eigenvectors of a non-normal matrix that best characterizes the one-step evolution of the measured vector [12], allowing for the data-driven discovery of fundamental transition dynamics. The development of DMD is timely due to the concurrent rise of data science, encompassing a broad range of techniques, from machine learning and statistical regression to computer vision and compressed sensing [10]. To the best of the authors' knowledge, this is the first attempt to apply the DMD algorithm on emotional speech.

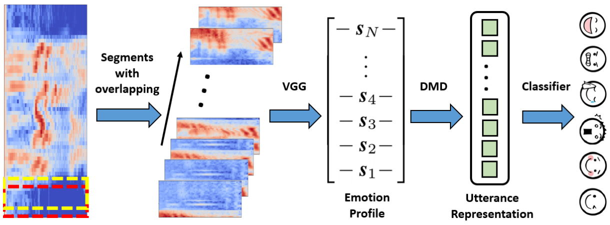

Figure 1: *Illustration of the proposed method*

Our method builds on the concept termed Emotion Distillation, which is the process of generating a set of emotion-specific features from the original high-dimensional feature space that explicitly describes the *emotion fluctuations* over time [13]. Distillation features are strongly tied to the task of interest, naturally highlighting salient portions of the data. In this work, we distill emotion information using Emotion Profiles (EPs) [13]. EPs are typically represented by a multi-dimensional signal, where each dimension represents a classifier-derived estimate of probability distribution of a set of basic emotion content (e. g., angry, fear, happy, sad). A large body of earlier works have demonstrated the efficacy of EPs in emotion-related tasks [13, 14, 15, 16]. This paper extends EPs into an end-to-end approach, where EPs are learned from log Mel spectrograms via a deep Convolutional Neural Network model. Furthermore, in contrast to the aforementioned earlier works, which only utilize the final estimates to constitute their EPs, this paper also investigates the bottleneck features to form EPs. Extensive experiments are conducted on two popular emotion corpora, namely, the CASIA corpus [17] and the SAVEE database [18]. Empirical results show the efficacy of the proposed method.

2. Methodology

Figure 1 illustrates the proposed framework. It comprises a VGG [19] deep Convolutional Neural Network (CNN) trained on log Mel filterbanks of individual segments to make the segment-level decision. The Emotion Profiles (EPs) are then generated and utilized for constructing utterance representations using the Dynamic Mode Decomposition (DMD) algorithm. Finally, a relatively simple classifier is employed to make the final decision.

http://dx.doi.org/10.21437/Interspeech.2020-1762

2.1. Emotion profiles (EPs)

Emotion Profiles (EPs) were introduced and demonstrated to be useful for emotion classification tasks in [13, 14, 15, 16]. Typically, EPs are time series estimates of a set of the typical "basic" emotions (e. g., angry, happy, neutral, sad), with each EP component estimating the degree of confidence in the presence or absence of the corresponding emotion cues across the utterance. We call this kind of EPs the *estimate-level* EPs (EEPs). In addition to EEPs, this work also explores the possibility of using bottleneck features for constructing EPs, called the *bottleneck feature-level* EPs (BEPs) in this paper. Many works [20, 21, 22] have shown that bottleneck features contain rich information. We herein posit that the BEPs might serve as a complementary feature source to the conventional EEPs.

2.1.1. Generating EPs

The EPs in this work are generated using a VGG model, which is trained on the 64-bin log Mel filterbanks of individual segments. The log Mel filterbanks are computed by *short-time Fourier transform* (STFT) with a window length of 25 ms, hop length of 10 ms, and FFT length of 512. Subsequently, 64-bin log Mel filterbank features are derived from each short-time frame, and the frame-level features are combined to form a time-frequency matrix representation of the segment. Each segment inherits the label of the utterance where it lies.

The trained VGG model aims to predict a probability distribution $\boldsymbol{P}_i$ for the i^{th} segment in a certain utterance:

$$\boldsymbol{P}_i = [p_i(e_1),\ p_i(e_2),\ \cdots,\ p_i(e_C)]^T \tag{1}$$

where, $e_1,\ e_2,\ \cdots,\ e_C$, represent the set of basic emotions. The EEP for a specific utterance $\boldsymbol{U}$ can then be expressed as

$$\boldsymbol{U}_{EEP} = [\boldsymbol{P}_1,\ \boldsymbol{P}_2,\ \cdots,\ \boldsymbol{P}_N] \tag{2}$$

Where N is the number of segments in the utterance.

Meanwhile, the outputs of the penultimate layer of the trained VGG, i. e., the *bottleneck features*, are utilized to construct the BEP for Utterance $\boldsymbol{U}$ as

$$\boldsymbol{U}_{BEP} = [\boldsymbol{B}_1,\ \boldsymbol{B}_2,\ \cdots,\ \boldsymbol{B}_N] \tag{3}$$

Where each $\boldsymbol{B}_i \in \mathbb{R}^{M\times 1}, i = 1, 2, .., N$, represents the bottleneck feature vector for the i^{th} segment in Utterance $\boldsymbol{U}$, with embedding dimension of M.

2.2. Dynamic Mode Decomposition (DMD)

For the purposes of applying the DMD method, the following matrix is first defined:

$$\boldsymbol{U}_j^k = [\boldsymbol{s}_j,\ \boldsymbol{s}_{j+1},\ \ldots,\ \boldsymbol{s}_k] \tag{4}$$

This matrix includes Segment j through k of Utterance $\boldsymbol{U}$. A segment, $\boldsymbol{s}_i$, can be replaced by a probability distribution $\boldsymbol{P}_i \in \mathbb{R}^{C\times 1}$ (for EEPs), or by a bottleneck feature vector $\boldsymbol{B}_i \in \mathbb{R}^{M\times 1}$ (for BEPs).

To construct the Koopman operator [10, 23] that best represents the data collected, the matrix $\boldsymbol{U}_1^N$ (i. e., the whole utterance) is considered:

$$\boldsymbol{U}_1^N = [\boldsymbol{s}_1,\ \boldsymbol{s}_2,\ \ldots,\ \boldsymbol{s}_N] \tag{5}$$

Where N is the number of segments in the utterance.

To apply standard DMD [9], the first-order Koopman assumption is employed:

$$\boldsymbol{s}_k = \boldsymbol{A}\boldsymbol{s}_{k-1} \tag{6}$$

The matrix $\boldsymbol{U}_1^N$ then reduces to

$$\boldsymbol{U}_1^N = [\boldsymbol{s}_1,\ \boldsymbol{A}\boldsymbol{s}_1,\ \ldots,\ \boldsymbol{A}^{N-1}\boldsymbol{s}_1] \tag{7}$$

or

$$\boldsymbol{U}_2^N = \boldsymbol{A}\boldsymbol{U}_1^{N-1} \tag{8}$$

Where $\boldsymbol{A}$ is the Koopman operator and is chosen to minimize the Frobenius norm of $||\boldsymbol{U}_2^N - \boldsymbol{A}\boldsymbol{U}_1^{N-1}||_F$. In other words, the operator $\boldsymbol{A}$ advances each segment in $\boldsymbol{U}_1^{N-1}$ a single time step, resulting in the corresponding future segments in $\boldsymbol{U}_2^N$. The operator $\boldsymbol{A}$ thus captures the overall transition dynamics of the utterance, and summarizing $\boldsymbol{A}$ would lead to the construction of the desired utterance representation.

The first-order Koopman assumption constrains a segment in an utterance to transition solely from the previous one. To make our assumption more realistic, we look towards the higher-order Koopman assumption [24]:

$$\boldsymbol{s}_k = \boldsymbol{A}_1\boldsymbol{s}_{k-1} + \cdots + \boldsymbol{A}_{d-1}\boldsymbol{s}_{k-d+1} + \boldsymbol{A}_d\boldsymbol{s}_{k-d} \tag{9}$$

Where d is the *order* parameter. This can be written in a form similar to Equation (5) and Equation (8), respectively:

$$\tilde{\boldsymbol{U}}_1^N = [\tilde{\boldsymbol{s}}_1,\ \tilde{\boldsymbol{s}}_2,\ \ldots,\ \tilde{\boldsymbol{s}}_N] \tag{10}$$

$$\tilde{\boldsymbol{U}}_2^N = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{U}}_1^{N-1} \tag{11}$$

where,

$$\tilde{\boldsymbol{s}}_k = [\boldsymbol{s}_k,\ \boldsymbol{s}_{k+1},\ \cdots,\ \boldsymbol{s}_{k+d-2},\ \boldsymbol{s}_{k+d-1}]^T \tag{12}$$

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{A}_d & \boldsymbol{A}_{d-1} & \boldsymbol{A}_{d-2} & \cdots & \boldsymbol{A}_2 & \boldsymbol{A}_1 \end{bmatrix} \tag{13}$$

with I being an identity matrix.

With this relaxation, a particular segment in an utterance is not only related to the preceding segment, but to several preceding segments with a window size of d, which is tunable, and $d = 1$ falls back to the first-order cases.

2.2.1. Constructing utterance representations

The higher-order Koopman operator $\tilde{\boldsymbol{A}}$ can be derived using Equation (11) as follows:

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}_2^N(\tilde{\boldsymbol{U}}_1^{N-1})^\dagger \tag{14}$$

where "†" denotes the pseudoinverse operation.

The dynamic modes and mode amplitudes can then be obtained by calculating the eigenvalues and eigenvectors of $\tilde{\boldsymbol{A}}$. In this paper, the dynamic mode (or eigenvector) that corresponds to the largest dynamic mode amplitude (or eigenvalue) is used as the utterance representation for the corresponding utterance, as it captures the largest-scale dynamic present in the sequence of segments. Algorithm 1 illustrates the overall process.

Algorithm 1 DMD algorithm for constructing an utterance representation

```
Input: (a) Sequence of segments in an utterance $\boldsymbol{U}_1^N = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N]$. (b) Order parameter $d$.
Outputs: Utterance representation.
1:  Declare $\tilde{\boldsymbol{U}}_1^N = [\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \ldots, \tilde{\boldsymbol{s}}_N]$, where $\tilde{\boldsymbol{s}}_k$ is given by Equation (12);
2:  Computing the higher-order Koopman operator $\tilde{\boldsymbol{A}}$:
        $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}_2^N (\tilde{\boldsymbol{U}}_1^{N-1})^\dagger$
3:  Performing eigendecomposition on $\tilde{\boldsymbol{A}}$:
        $[\boldsymbol{W}, \boldsymbol{D}] = eig(\tilde{\boldsymbol{A}})$
    where $\boldsymbol{D}$ is a diagonal matrix composed of sorted eigenvalues and the columns of matrix $\boldsymbol{W}$ are the corresponding right eigenvectors;
4: The top eigenvector in $\boldsymbol{W}$ that corresponds to the largest eigenvalue in $\boldsymbol{D}$ is selected for constructing the utterance representation.
```

2.3. Competing Methods

2.3.1. P-means

P-means [25] is a method that concatenates different types of means, also known as power-means [26]. The hypothesis is that the average is only one type of order-statistic, and there are several others available, which might add useful information when constructing utterance representations.

2.3.2. Functionals

The comparison to functionals is only natural, as it is a common practice within this community. The functionals employed in this work include arithmetic mean, Percentile 1, Percentile 99, and Quartiles 1-3.

2.3.3. Discrete Cosine Transform

The Discrete Cosine Transform (DCT) algorithm is widely used in digital signal processing applications for summarizing or compressing information. In this paper, DCT is applied on the EPs. Taking the BEPs, for example, given an utterance of N segments $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N$, we stack the sequence of M-dimensional BEPs into an $N \times M$ matrix. The DCT algorithm is then applied along the M columns, respectively. To get a fixed-length utterance representation, we extract and concatenate the first K DCT coefficients and discard higher-order coefficients, which results in consistent utterance vectors of size KM. For cases where $N < K$, we pad the utterance with K - N zero vectors.

3. Emotion Corpora

Two different emotion corpora are used to evaluate the validity and universality of our approach, i. e., a Chinese emotional corpus (CASIA) [17] and an English emotional database (SAVEE) [18]. All of the emotion categories are selected for each of the two emotion corpora, respectively.

Specifically, the CASIA corpus [17] contains 9,600 utterances that are simulated by four subjects (two males and two females) in six different emotional states, i. e., angry, fear, happy, neutral, sad, and surprise. In our experiments, we only use 7,200 utterances that correspond to 300 linguistically neutral sentences with the same statements.

The Surrey audio-visual expressed emotion database (SAVEE) [18] consists of recordings from four male actors in seven different emotions: anger, disgust, fear, happy, sad, surprise, and neutral. Each speaker produced 120 utterances. The sentences were chosen from the standard TIMIT corpus and phonetically-balanced for each emotion.

4. Experiments

4.1. Setup

According to [27, 16], a speech segment longer than 250 ms contains sufficient emotional information to identify the emotion being expressed in that segment. In our experiment, the size of each speech segment is set to 32 frames, i.e., the total length of a segment is 10 ms × 32 + (25 - 10) ms = 335 ms. For the CASIA corpus, the segment hop length is set to 30 ms, while it is set to 10 ms for the SAVEE database. In this way, we collected 418,722 segments for the CASIA corpus and 51,027 segments for the SAVEE database.

For the VGG network, the architecture of the convolutional layers is based on the configurations (i. e., configuration E) in the original paper [19]. the only change we made was to the last three FC layers ($\{128, 32, C\}$ units, respectively, with C denoting the number of possible emotions). In the training stage, ADAM [28] optimizer with default setting in Tensorflow [29] was used, with an initial learning rate of 0.001 and an exponential decay scheme with a rate of 0.8 every two epochs. The batch size was set to 128. Early stopping with patience of 3 epochs was utilized to mitigate an overfitting problem.

The EPs are generated using leave-one-fold-out ten-fold cross-validation. A *random forest* (RF) with default setting in Scikit-learn [30] was then employed to make the utterance-level decision, where another ten-fold cross-validation is performed. The results are presented in terms of unweighted accuracy (UA) and weighted accuracy (WA), respectively. It is worth noting that the UA and WA are the same for the CASIA corpus as it is perfectly balanced concerning the emotion category.

4.2. Results and analysis

Table 1-3 show the results of the experiments performed with P-means, DCT, and DMD on the two stated emotional corpora, respectively. The following can be seen: (1) P-means achieved impressive performance, indicating the importance of the scale information (e. g., the average) of an emotional speech utterance. Also, adding higher-order powers was beneficial overall, which corroborated our previous hypothesis. (2) Both of DCT and DMD methods did achieve respectable results, demonstrating that the dynamic information plays an essential role in characterizing the emotional speech as well. (3) Based on the results of the DCT method in Table 2, it can be seen that the DCT method needs more components to keep for a relatively large database (CASIA) than a small one (SAVEE), to achieve reasonable performance. (4) Observing the results of the DMD method in Table 3, it is clear that exploiting the higher-order assumption (see Equation 9) is beneficial for the relatively large database (CASIA), since the results are better for adding higher values of the order parameter. (5) P-means outperformed both DCT and DMD-based techniques. We thus posit that the scale information is more critical than dynamics in this task.

The summary of results is provided in Table 4, where the best results for each method are provided. It also has an additional result where the most performant EigenEmo-based utter-

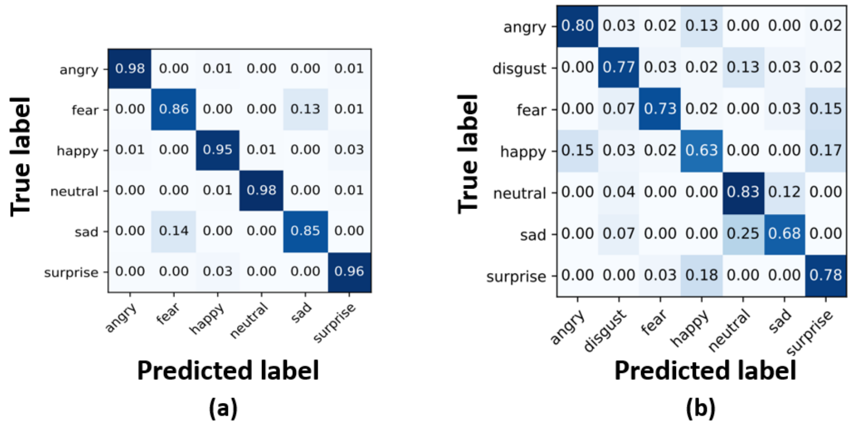

Figure 2: *Confusion matrices obtained using the most performant EigenEmo-based utterance representations for (a) CASIA corpus, where the "DMD⊕AVG & EEP⊕BEP" method was applied, and (b) SAVEE database, where the "DMD⊕AVG & EEP" method was applied, respectively (refer to Table 4).*

Table 1: *Results with p-means on the two selected corpora. The Power component is varied between 1, [1-2], [1-3] and [1-6].*

P-means	CASIA				SAVEE			
	EEP		BEP		EEP		BEP	
Power(s)	WA	UA	WA	UA	WA	UA	WA	UA
1	92.01	92.01	91.53	91.53	73.33	71.90	**68.13**	**65.60**
[1-2]	**92.40**	**92.40**	92.11	92.11	73.75	**72.38**	65.83	62.98
[1-3]	92.19	91.19	**92.25**	**92.25**	72.50	70.83	63.96	60.60
[1-6]	92.17	92.17	92.15	92.15	**73.96**	**72.38**	61.88	58.21

Table 2: *Results with DCT on the two selected corpora. The number of DCT Components (Cmp.) is varied from* 1 *to* 6.

DCT	CASIA				SAVEE			
	EEP		BEP		EEP		BEP	
Cmp.	WA	UA	WA	UA	WA	UA	WA	UA
1	89.68	89.68	90.15	90.15	**67.08**	**63.45**	**62.08**	**58.57**
2	89.53	89.53	89.78	89.78	61.25	57.38	59.17	54.64
3	**90.04**	**90.04**	90.63	90.63	61.67	57.38	54.17	48.69
4	89.82	89.82	90.49	90.49	63.33	59.40	54.38	49.17
5	89.58	89.58	90.56	90.56	63.54	59.64	52.08	46.07
6	89.67	89.67	**90.69**	**90.69**	64.38	60.36	51.46	45.24

Table 3: *Results with DMD on the two selected corpora. d is the window size as described in Equation (9) and is varied between 1, 2, 3, 6, [1-2], [1-3] and [1-6].*

DMD	CASIA				SAVEE			
	EEP		BEP		EEP		BEP	
d	WA	UA	WA	UA	WA	UA	WA	UA
1	90.71	90.71	90.50	90.50	**73.96**	**72.55**	**63.75**	**60.83**
2	90.65	90.65	91.03	91.03	73.13	71.48	62.50	59.29
3	90.04	90.04	90.25	90.25	72.71	70.95	62.08	58.81
6	89.88	89.88	89.71	89.71	71.25	69.29	61.67	58.52
[1-2]	90.83	90.83	**91.50**	**91.50**	73.54	71.95	61.25	58.10
[1-3]	91.06	91.06	91.29	91.29	72.08	70.71	61.88	58.45
[1-6]	**91.33**	**91.33**	91.07	91.07	71.04	69.40	61.04	57.86

ance representations have been concatenated with the averaged EPs. It can be readily seen that this concatenation significantly improved performance, as the resulting representation can now capture both the scale and dynamics of an emotional speech utterance. Figure 2 shows the corresponding confusion matrices.

Table 4: *Comparison of methods on the two selected corpora. "⊕" means features are concatenated.*

		CASIA		SAVEE	
Method	**EP Type**	WA	UA	WA	UA
P-means	EEP	92.40	92.40	73.96	72.38
P-means	BEP	92.25	92.25	68.13	65.60
P-means	EEP⊕BEP	92.33	92.33	73.13	71.69
DCT	EEP	90.04	90.04	67.08	63.45
DCT	BEP	90.69	90.69	62.08	58.57
DCT	EEP⊕BEP	91.10	91.10	65.83	63.67
Functionals	EEP	92.53	92.53	74.15	73.03
Functionals	BEP	92.36	92.36	64.79	62.52
Functionals	EEP⊕BEP	92.47	92.47	73.96	72.55
DMD	EEP	91.33	91.33	73.96	72.55
DMD	BEP	91.50	91.50	63.75	60.83
DMD	EEP⊕BEP	92.07	92.07	71.46	70.60
DMD⊕AVG	EEP	92.04	92.04	**75.83**	**74.76**
DMD⊕AVG	BEP	92.50	92.50	68.13	67.33
DMD⊕AVG	EEP⊕BEP	**93.28**	**93.28**	73.33	71.38

5. Conclusions

In this paper, we proposed a novel method to construct utterance representation for speech emotion classification by exploiting the dynamic properties of the emotion profiles generated by a VGG network. We do this using a spectral decomposition method rooted in fluid-dynamics, known as Dynamic Mode Decomposition. Empirical validation of the proposed method on the CASIA corpus and the SAVEE database shows promising results. Since we herein blindly used all segments to train the segment-level classifier, it is anticipated with proper segment selection strategy, better results are expected.

6. References

[1] B. Schuller, G. Rigoll, and M. Lang, "Hidden Markov model-based speech emotion recognition," in *Proc. ICASSP*, vol. 2, 2003, pp. II–1.

[2] B. Vlasenko, D. Prylipko, R. Böck, and A. Wendemuth, "Modeling phonetic pattern variability in favor of the creation of robust emotion classifiers for real-life applications," *Computer Speech & Language*, vol. 28, pp. 483–500, 2014.

[3] B. Vlasenko, "Emotion recognition within spoken dialog systems," *PhD thesis. University of Magdeburg*, 2011.

[4] M. Wöllmer, A. Metallinou, N. Katsamanis, B. Schuller, and S. Narayanan, "Analyzing the memory of blstm neural networks for enhanced emotion classification in dyadic spoken interactions," in *Proc. ICASSP*, 2012, pp. 4157–4160.

[5] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proc. INTERSPEECH*, 2015.

[6] C. Huang and S. S. Narayanan, "Attention assisted discovery of sub-utterance structure in speech emotion recognition." in *Proc. INTERSPEECH*, 2016, pp. 1387–1391.

[7] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. ICASSP*, 2017, pp. 2227–2231.

[8] W. Han, H. Ruan, X. Chen, Z. Wang, H. Li, and B. Schuller, "Towards temporal modelling of categorical speech emotion recognition," in *Proc. INTERSPEECH*, 2018, pp. 932–936.

[9] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[10] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[11] S. Kayal and G. Tsatsaronis, "Eigensent: Spectral sentence embeddings using higher-order dynamic mode decomposition," in *Proc. ACL*, 2019, pp. 4536–4546.

[12] A. Prasadan and R. R. Nadakuditi, "Time series source separation using dynamic mode decomposition," *arXiv preprint arXiv:1903.01310*, 2019.

[13] E. M. Provost and S. Narayanan, "Simplifying emotion classification through emotion distillation," in *Proc. APSIPA*, 2012, pp. 1–4.

[14] E. Mower, M. J. Mataric, and S. Narayanan, "A framework for automatic human emotion classification using emotion profiles," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1057–1070, 2010.

[15] Y. Shangguan and E. M. Provost, "Emoshapelets: Capturing local dynamics of audio-visual affective speech," in *Proc. ACII*, 2015, pp. 229–235.

[16] Y. Kim and E. M. Provost, "Emotion classification via utterance-level dynamics: A pattern-based approach to characterizing affective expressions," in *Proc. ICASSP*, 2013, pp. 3677–3681.

[17] J. Tao, F. Liu, M. Zhang, and H. Jia, "Design of speech corpus for mandarin text to speech," in *Proc. the 4th Workshop on Blizzard Challenge*, 2005.

[18] P. Jackson and S. Haq, "Surrey audio-visual expressed emotion (savee) database," *University of Surrey: Guildford, UK*, 2014.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE signal processing letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[21] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," in *Proc. DCASE*, 2018.

[22] X. Meng, B. Leng, and G. Song, "A multi-level weighted representation for person re-identification," in *Proc. ICANN*, 2017, pp. 80–88.

[23] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the national academy of sciences of the united states of america*, vol. 17, no. 5, p. 315, 1931.

[24] S. Le Clainche and J. M. Vega, "Higher order dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 882–925, 2017.

[25] A. Rücklé, S. Eger, M. Peyrard, and I. Gurevych, "Concatenated power mean embeddings as universal cross-lingual sentence representations," *arXiv preprint arXiv:1803.01400*, 2018.

[26] G. H. Hardy, J. E. Littlewood, G. Pólya, G. Pólya, D. Littlewood *et al.*, *Inequalities*. Cambridge university press, 1952.

[27] E. M. Provost, "Identifying salient sub-utterance emotion dynamics using flexible units and estimates of affective flow," in *Proc. ICASSP*, 2013, pp. 3682–3686.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, 2016, pp. 265–283.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.