openEAR - Introducing the Munich Open-Source Emotion and Affect Recognition Toolkit

Florian Eyben, Martin Wöllmer, and Björn Schuller
Technische Universität München, Institute for Human-Machine Communication
Theresienstrasse 90, 80333 München
{eyben|woellmer|schuller}@tum.de

Abstract

Various open-source toolkits exist for speech recognition and speech processing. These toolkits have brought a great benefit to the research community, i.e. speeding up research. Yet, no such freely available toolkit exists for automatic affect recognition from speech. We herein introduce a novel open-source affect and emotion recognition engine, which integrates all necessary components in one highly efficient software package. The components include audio recording and audio file reading, state-of-the-art paralinguistic feature extraction and plugable classification modules. In this paper we introduce the engine and extensive baseline results. Pre-trained models for four affect recognition tasks are included in the openEAR distribution. The engine is tailored for multi-threaded, incremental on-line processing of live input in real-time, however it can also be used for batch processing of databases.

1. Introduction

Affective Computing has become a popular area of research in recent times [17]. Many achievements have been made towards making machines detect and understand human affective states, such as emotion, interest or dialogue role. Yet, in contrast to the field of speech recognition, only very few software toolkits exist, which are tailored specifically for affect recognition from audio or video. In this paper, we introduce and describe the Munich open Affect Recognition Toolkit (openEAR), the first such tool, which runs on multiple platforms and is publicly available[1].

OpenEAR in it's initial version is introduced as an affect and emotion recognition toolkit for audio and speech affect recognition. However, openEAR's architecture is modular and by principle modality independent. Thus, also vision features such as facial points or optical flow measures can be added and fused with audio features. Moreover, physiological features such as heart rate, ECG, or EEG signals from devices such as the Neural Impulse Actuator (NIA), can be analysed using the same methods and algorithms as for speech signals and thus can also be processed using openEAR – provided suitable capture interfaces and databases.

2. Existing work

A few free toolkits exist, that provide various components usable for emotion recognition. Most toolkits that include feature extraction algorithms are targeted at speech recognition and speech processing, such as the Hidden Markov Toolkit (HTK) [16], the PRAAT Software [1], the Speech Filling System (SFS) from UCL, and the SNACK package for the Tcl scripting language. These can all be used to extract state-of-the-art features for emotion recognition. However, only PRAAT and HTK include certain classifiers. For further classifiers WEKA and RapidMiner, for example, can be used. Moreover, only few of the listed toolkits are available under a permissive Open-Source license, e. g. WEKA, PRAAT, and RapidMiner.

The most complete and task specific framework for Emotion Recognition currently is EmoVoice [13]. However, the main design objective is to provide an emotion recognition system for the non-expert. Thus it is a great framework for demonstrator applications and making emotion recognition available to the non-expert. openEAR, in contrast, aims at being a stable and efficient set of tools for researchers and those developing emotional aware applications, providing the elementary functionality for emotion recognition, i. e. the Swiss Army Knife for research and development of affect aware applications. openEAR combines everything from audio recording, feature extraction, and classification to evaluation of results, and pre-trained models while being very fast and highly efficient. All feature extractor components are written in C++ and can be used as a library, facilitating integration into custom appli-

[1] http://sourceforge.net/projects/openear

978-1-4244-4799-2/09/$25.00

cations. Also, openEAR can be used as an out-of-the-box emotion live affect recogniser for various domains, using pre-trained models which are included in the distribution. Moreover, openEAR is Open-Source software, freely available to anybody under the terms of the GNU General Public License.

3. openEAR's Architecture

The openEAR toolkit consists of three major components: the core component is the SMILE (Speech and Music Interpretation by Large-Space Extraction) signal processing and feature extraction tool, which is capable generating $> 500\,k$ features in real-time (Real-Time Factor (RTF) < 0.1), either from live audio input or from off-line media. Next, there is support for classification modules via a plug-in interface to the feature extractor. Moreover, supporting scripts and tools are provided, which facilitate training of own models on arbitrary data sets. Finally, four ready-to-use model-sets are provided for recognition of six basic emotion categories (trained on the Berlin Speech Emotion Database (EMO-DB) [2] and the eNTERFACE database), for recognition of emotion in a continuous three-dimensional feature space spanned by activation, valence, and dominance (trained on the Belfast naturalistic (SAL) and Vera-am-Mittag (VAM) [4] corpora), for recognition of interest using three discrete classes taken from the Audio Visual Interest Corpus (AVIC) [9], and for recognition of affective states such as drunkenness trained on the Airplane Behaviour Corpus (ABC).

Signal input can either be read off-line from audio files or recorded on-line from a capture device in real-time. Since data processing is incremental (concerning signal processing and feature extraction), there is no difference between handling live input and off-line media. Independent of the input method, the feature output can either be classified directly via built in classifiers, classifier plug-ins, or the features (or even wave data) can be exported to various file formats used by other popular toolkits. Currently implemented export file formats are: WEKA Arff [14], LibSVM format [3], Comma Separated Value (CSV) File, and Hidden Markov Toolkit (HTK) [16] feature files.

The following sub-sections describe the feature extractor's modular architecture, the features currently implemented, and the classifier interface. The model-sets will be detailed along with baseline benchmark results in section 4.

3.1. Modular and Efficient Implementation

During specification of openEAR's feature extractor architecture, three main objectives were followed: speed and efficiency, incremental processing of data (i. e. frame by frame with minimum delay), and flexibility and modularity. Adding new features is possible via an easy plug-in interface.

The SMILE feature extractor is implemented from scratch in C++, without crucial third party dependencies. Thus, it is easy to compile, and basically platform independent. It is currently known to run on Mac OS, various Linux distributions, and Windows platforms. Feature extraction code is optimised to avoid double computations of shared values, e. g. Fast Fourier Transform (FFT) coefficients, which are only computed once and used for multiple algorithms such as computation of energy, spectral features, and cepstral features.

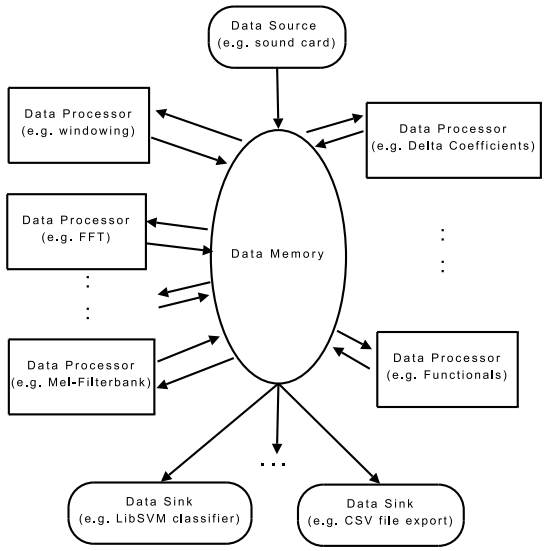

Figure 1. Concept and components of openEAR's SMILE (Speech and Music Interpretation by Large-Space Extraction) feature extractor.

Figure 1 shows a rough sketch of the data flow and signal processing architecture. The central component is the *Data Memory*, which enables memory efficient incremental processing by managing ring-buffer storage of feature data. Input data (wave files, other features, etc.) is fed to the *Data Memory* by *Data Source* components, which contain a *Data Writer* sub-component that handles the data memory interface. *Data Processor* components read data frames or contours from one location of the *Data Memory*, process the data and write new frames to a different location in the *Data Memory*. They contain both a *Data Reader* and a *Data Writer* sub-component, which handle the *Data Memory* interface. Finally, the *Data Sink* components read data from the *Data Memory* and feed it to the classifier components or write data to files. Each component can be run in a separate thread, speeding up processing on multiple processors/cores.

The individual components can be freely instantiated, configured, and connected to the *Data Memory* via a central configuration file. To facilitate configuration file creation example files are provided and configuration file conversion

scripts are included.

3.2. Features

The SMILE feature extraction tool is capable of extracting low-level audio features (Low-Level Descriptors (LLD)) and applying various statistical functionals and transformations to those features. The Low-Level Descriptors currently implemented are listed in table 1. We hope to extend this list by numerous advanced and state-of-the-art features in the near future, such as Voice Quality Parameters (e.g. [6]), alternative pitch detection algorithms, e. g. pitch by Harmonic Product Spectrum, combination of Average Magnitude Difference with Autocorrelation, and smoothing of pitch contours via a Viterbi algorithm or ESPS pitch tracker. Moreover, features such as TEAGER energy or further Auditory Features are considered for integration.

Feature Group	**Features in Group**
Signal energy	Root mean-square & logarithmic
FFT-Spectrum	Bins 0-N_{fft}
Mel-Spectrum	Bins 0-N_{mel}
Cepstral	MFCC 0-N_{mfcc}
Pitch	Fundamental frequency F_0 via ACF, in Hz, Bark and closest semitone.
	Probability of voicing ($\frac{ACF(T_0)}{ACF(0)}$)
Voice Quality	Harmonics-to-noise ratio
LPC	LPC Coefficients
PLP	Perceptual Linear Predictive Coefficients
Formants	Formants and Bandwidth computed from LPC analysis
Time Signal	Zero-crossing-rate, maximum value, minimum value, DC
Spectral	Energy in bands 0 - 250 Hz, 0 - 650 Hz, 250 - 650 Hz, 1 - 4 kHz, and custom
	N % roll-off point, centroid, flux, and rel. pos. of spectrum max. and min.
Musical	CHROMA (warped semitone filterbank), CENS
	Comb-filter bank

Table 1. Low-Level Descriptors implemented in openEAR's SMILE feature extractor.

The Mel frequency features, Mel-Spectrum and Mel-Frequency Cepstral Coefficients (MFCCs) are computed exactly as described in [16], thus providing compatibility to the Hidden Markov Toolkit and existing models trained on HTK MFCCs. Harmonics-To-Noise Ratio computation is based on equation 1, where T_0 is the pitch period.

$$HNR^t = 10 \cdot \log \frac{ACF(T_0)}{ACF(0) - ACF(T_0)} \tag{1}$$

Spectral centroid (C_S^t) at time t is computed via equation 2. $X^t(f)$ is the spectral magnitude at time t in bin f.

$$C_S^t = \frac{\sum_{\forall f} f \cdot X^t(f)}{\cdot \sum_{\forall f} X^t(f)} \tag{2}$$

Spectral Flux F_S^t for N FFT bins is computed via equation 3, whereby E^t is the energy of the frame at time t.

$$F_S^t = \sqrt{\frac{1}{N} \sum_{f=1}^{N} \left(\frac{X^t(f)}{E^t} - \frac{X^{t-1}(f)}{E^{t-1}} \right)^2} \tag{3}$$

The p percent Spectral Roll-Off is determined as the frequency (or FFT bin) below which p percent of the total signal energy fall. All frequencies, i. e. for Centroid and Roll-Offs are normalised to 1000 Hz.

Delta regression coefficients d^t of arbitrary order can be computed from any LLD contour (x^t) using equation 4, where W specifies half the size of the window to be used for computation of the regression coefficients. The default is $W = 2$. In order to provide HTK compatibility, equation 4 was implemented as described in [16]. Typically only the first and second order δ-coefficients are used, which is also the default setting in openEAR.

$$d^t = \frac{\sum_{i=1}^{W} i \cdot \left(x^{t+i} - x^{t-i} \right)}{2 \sum_{i=1}^{W} i^2} \tag{4}$$

Table 2 lists the statistical functionals, regression coefficients and transformations currently implemented. They can be applied to the LLD to map a time sequence of variable length to a static feature vector. This procedure is the common technique in related emotion and affect recognition work (cf. [7]). Moreover, hierarchical functionals can be computed as "functionals of functionals" (cf. [11]), which helps improve robustness against single outliers, for example. Thereby there is no limit as to how many hierarchies can be computed, except by computing resources such as memory and processing time.

As with the LLD we aim at implementing even more functionals which can be applied to LLD contours or functional contours, in order to facilitate systematic feature generation. Due to the modular architecture of the feature extractor, it will also be possible to apply any implemented processing algorithm to any time series, i. e. the Mel-band filter-bank could be applied as a functional to any LLD contour. This gives researchers an efficient and customisable tool to generate millions of features in order to find optimal feature sets which represent affective information.

Functionals, etc.	#
Max./min. val. and respective rel. position	4
Range (max.-min.)	1
Arithmetic, quadratic, and geometric mean	3
Arth. mean of absând non-zero val.	2
%-age of non-zero val. wrt. tot. # of val. in contour	1
Max. and min. value - arithmetic mean	2
Quartiles and inter-quartile ranges	6
N % percentiles	(N)
Std. deviation, variance, kurtosis, skewness	4
Centroid of LLD contour	1
Zero-crossing and mean-crossing rate	2
25% Down-Level Time, 75% Up-Level Time, Rise-Time, Fall-Time	4
# of peaks, mean dist. btwn. peaks, arth. mean of peaks, arth. mean of peaks - overall arth. mean	4
Number of segments based on δ-thresholding	1
Linear reg. coefficients and corresp. approx. err.	4
Quadratic reg. coefficients and corresp. approx. err.	5
Discrete Cosine Transf. (DCT) coefficients 0-N	(6)

Table 2. Statistical functionals, regression coefficients and transformations currently implemented in openEAR's feature extractor.

3.3. Classification and Data Output

As mentioned in section 3.1, multiple *Data Sinks* can be present in the feature extractor, feeding data to different classifiers. There exists an implementation of a K-Nearest Neighbour classifier, a Bayes classifier, and a module for Support-Vector classification and regression using the efficient and freely available LibSVM [3]. We further plan to implement Discriminant Multinomial Naive Bayes [12] and Long Short-Term Memory Recurrent Neural Networks [5].

Supporting scripts written in Perl (in order to be platform independent) facilitate batch processing of data-sets. As features can be saved in various data-formats, custom experiments can easily be conducted using e. g. WEKA [14] or HTK [16]. Calling of various WEKA functions from the command-line, e. g. for feature selection or classification, is also included among openEAR's Perl scripts.

A benefit of openEAR's modular feature extractor architecture is that practically any internal data can be accessed and output to various formats simply by connecting a data sink to the specific location in the data memory. The exported data can be used for visualisation purposes, for example. This is helpful in finding extractor parametrisation problems, or for common sense checks to see if everything is working as expected. Visualising the internal data can also be a helpful means for teaching and understanding the process of extracting various feature types. Exemplary plots of some selected LLD can be found in figure 3.3.

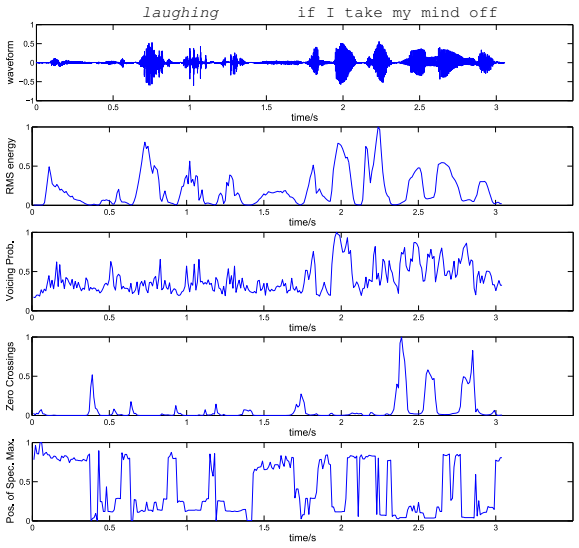

Figure 2. Plots of selected LLD for the spoken utterance: "*laughing*, if I take my mind off ..." (SAL database, male speaker). Top to bottom: original wave (3s), norm. RMS energy, probability of voicing, norm. zero-crossing rate, rel. freq. of spec. max.

4. Benchmarks

We now show benchmark results obtained on popular databases for speech emotion and interest recognition. Freely available models for all databases used for benchmarking are distributed with openEAR. Moreover, computation time performance of openEAR's SMILE feature extractor is evaluated. The results show, that openEAR yields state-of-the-art recognition results with very low computational demand.

The following sub-section briefly describes the databases, before the results are discussed in section 4.2.

4.1. Databases

Six databases as listed in table 3 are used for benchmarking openEAR and for which freely available model sets are distributed with openEAR: the first three contain discrete class labels for emotion, namely the Berlin Speech Emotion Database (EMO-DB) [2], containing seven classes of basic emotions (Anger, Fear, Happiness, Disgust, Boredom, Sadness, Neutral), the eNTERFACE corpus with six emotion categories (Anger, Disgust, Fear, Happiness, Sadness, and Surprise), the ABC corpus with the classes (Aggressive, Cheerful, Intoxicated, Nervous, Neutral, Tired), and the Audio Visual Interest Corpus (AVIC) [9] with labels for three levels of interest (-1: disinterest, 0: normal, and 1: high interest). The last two databases contain continuous dimensional labels for valence and activation in the range from -1 to +1. These are the SAL corpus and the VAM corpus [4]. The latter also has labels for the potency di-

mension. However, we found this dimension to be highly correlated to activation (Corr. Coeff. 0.9). Thus, we did not consider it in the evaluations. When viewing the results in

Database	# turns	discrete	continuous
ABC	431	✓	
AVIC	996	✓	
EMO-DB	494	✓	
eNTERFACE	1170	✓	
SAL	1692		✓
VAM	947		✓

Table 3. Six databases for benchmarking openEAR and generation of model sets included in the openEAR distribution.

the following section it has to be considered that ABC, eNTERFACE, and EMO-DB contain acted and prototypical emotions, while AVIC, SAL, and VAM contain natural and spontaneous data. Due to the dimensional annotation for SAL and VAM all recorded turns are left in the database, not only prototypical turns.

4.2. Recognition Results

Table 4 shows results obtained for discrete class emotion and interest (on AVIC) recognition. For all benchmarks the feature set *5,967 tF* as described in table 6 was extracted. A correlation based feature subset (CFS) selection was performed in order to find relevant feature sets for each database. However, better results with the full feature set were achieved on EMO-DB, eNTERFACE, and ABC. Thus, the best results without feature selection are shown there. All the benchmark results are well in line with or even above the current state-of-the-art.

Recall [%]	WA	UA
ABC (6 emo. rel. states)	71.9	66.5
AVIC (3 levels of interest)	74.5	70.4
EMO-DB (7 emotions)	89.5	88.8
eNTERFACE (6 emotions)	75.2	75.1

Table 4. Results obtained for discrete class emotion recognition using Support-Vector Machines with polynomial kernel function of degree 1. 10-fold Stratified Cross-Validation. Weighted average (WA) and unweighted average (UA) of class-wise recall rates as demanded in [10].

Finally, table 5 shows the results for the two most challenging tasks, the dimensional estimation of naturalistic emotions. Results obtained for VAM are slightly better than those reported for polynomial kernel SVM in [4]. Results for SAL are obtained on the same data-set partitions as in [15].

Database	CC_a	MLE_a	CC_v	MLE_v
SAL (train/test)	0.24	0.28	0.15	0.38
VAM (10-f. SCV)	0.83	0.15	0.42	0.14

Table 5. Results obtained for continuous emotion recognition for two dimensions (activation and valence) using Support-Vector Regression (polynomial kernel function, degree 1). Results reported are Correlation Coefficient (CC) and Mean absolute (linear) error (MLE) for activation (a) and valence (v). VAM: 10-fold Stratified Cross-Validation (SCV) after CFS feature selection. SAL: pre-defined train/test sets, CFS feature selection on training set.

Feature Set	Description
36 MFCCde	MFCC 0-12, first and second order δ
102 LLD	LLD pitch, time, spectral, mfcc, and energy + first and second order δ
5,031 tF	43 functionals (applied to complete input) of 39 LLD + first and second order δ. (No percentile functionals)
5,967 tF	51 functionals (applied to complete input) of 39 LLD + first and second order δ
5,031 2sF	43 functionals (applied to 2s windows w. 50% overlap) of 39 LLD + first and second order δ. (No percentile functionals)
5,967 2sF	51 functionals (applied to 2s windows w. 50% overlap) of 39 LLD + first and second order δ
216k tHRF	43 functionals applied to set *5,031 2sF*
304k tHRF	51 functionals applied to set *5,967 2sF*

Table 6. Feature-sets for openEAR feature extractor computation time evaluation and benchmark results.

4.3. Computation Time

Since one objective of openEAR is efficiency and real-time operation, we now provide run-time benchmarks for various feature sets, which are summarised in table 6. Computation time is evaluated under Ubuntu Linux on a AMD Phenom 64 bit CPU at 2.2GHz. All components are run in a single thread for benchmarking.

Table 7 shows the computation time and the Real-Time Factor (RTF) for extraction of various feature sets.

5. Conclusion and Outlook

We introduced openEAR, an efficient, open-source, multi-threaded, real-time emotion recognition framework providing an extensible, platform independent feature extractor implemented in C++, pre-trained models on six databases which are ready-to-use for on-line emotion and affect recognition, and supporting scripts for model building, evaluation, and visualisation. The framework is com-

Feature Set	Comp. time [s]	RTF
36 MFCCde	1.3	0.003
102 LLD	4.4	0.009
5,031 tF	6.1	0.012
5,031 2sF	7.2	0.014
216k tHRF	9.2	0.018

Table 7. openEAR's computation time and real-time factor (RTF) for feature extraction of 8 minutes and 27 seconds 16-bit mono audio sampled at 16 kHz. LLD for all above feature sets computed for 25 ms frames at a rate of 10 ms. CPU: AMD Phenom, 2.2 GHz. Single thread processing.

patible with related tool-kits, such as HTK and WEKA by supporting their data-formats. The current implementation was successfully evaluated on six affective speech corpora, showing state-of-the-art performance. Moreover, features for the Interspeech 2009 Emotion Challenge [10] were extracted with openEAR.

Development of openEAR is still in progress and more features will be added soon. Due to it's modular architecture and the public source code, rapid addition of new, advanced features by the community is hopefully encouraged.

Although openEAR already is a fully featured emotion and affect recognition toolkit, it can also be used for other tasks such as classification of non-linguistic vocalisations [8]. In the future, decoders for continuous speech recognition and linguistic features will be integrated into openEAR, resulting in a highly efficient and comprehensive affect recognition engine.

6. Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211486 (SEMAINE).

References

[1] P. Boersma and D. Weenink. Praat: doing phonetics by computer (version 4.3.14). `http://www.praat.org/`, 2005.

[2] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss. A database of german emotional speech. In *Proceedings Interspeech 2005, Lissabon, Portugal*, pages 1517–1520, 2005.

[3] C.-C. Chang and C.-J. Lin. *LibSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] M. Grimm, K. Kroschel, and S. Narayanan. Support vector regression for automatic recognition of spontaneous emotions in speech. In *International Conference on Acoustics, Speech and Signal Processing, 2007.*, volume 4, pages IV–1085–IV. IEEE, April 2007.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[6] M. Lugger and B. Yang. Psychological motivated multi-stage emotion classification exploiting voice quality features. In F. Mihelic and J. Zibert, editors, *Speech Recognition*, page 1. IN-TECH, November 2008.

[7] B. Schuller, A. Batliner, D. Seppi, S. Steidl, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, N. Amir, L. Kessous, and V. Aharonson. The relevance of feature type for the automatic classification of emotional user states: Low level descriptors and functionals. In *Proc. INTERSPEECH 2007*, pages 2253–2256, Antwerp, Belgium, 2007.

[8] B. Schuller, F. Eyben, and G. Rigoll. Static and dynamic modelling for the recognition of non-verbal vocalisations in conversational speech. In E. André, editor, *Proc. IEEE PIT 2008*, volume LNCS 5078, pages 99–110. Springer, 2008. 16.-18.06.2008.

[9] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Höthker, and H. Konosu. Being bored? recognising natural interest by extensive audiovisual integration for real-life application. *to appear in Image and Vision Computing Journal (IMAVIS), Special Issue on Visual and Multimodal Analysis of Human Spontaneous Behavior, Elsevier*, page 17 pages, 2009.

[10] B. Schuller, S. Steidl, and A. Batliner. The interspeech 2009 emotion challenge. In *Interspeech (2009), ISCA, Brighton, UK*, 2009.

[11] B. Schuller, M. Wimmer, L. Mösenlechner, C. Kern, D. Arsic, and G. Rigoll. Brute-forcing hierarchical functionals for paralinguistics: A waste of feature space? In *Proceedings of ICASSP 2008*, Las Vegas, Nevada, USA, April 2008.

[12] J. Su, H. Zhang, C. X. Ling, and S. Matwin. Discriminative parameter learning for bayesian networks. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1016–1023, New York, NY, USA, 2008. ACM.

[13] T. Vogt, E. André, and N. Bee. Emovoice - a framework for online recognition of emotions from voice. In *Proc. IEEE PIT 2008*, volume 5078 of *LNCS*, pages 188–199. Springer, June 2008. feature extraction, emotion recognition, GUI.

[14] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.

[15] M. Wöllmer, F. Eyben, S. Reiter, B. Schuller, C. Cox, E. Douglas-Cowie, and R. Cowie. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *Proceedings Interspeech*, Brisbane, Australia, 2008.

[16] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK book (v3.4)*. Cambridge University Press, Cambridge, UK, December 2006.

[17] Z. Zeng, M. Pantic, G. I. Rosiman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Trans. on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.