HIERARCHICAL NEURAL NETWORKS AND ENHANCED CLASS POSTERIORS FOR SOCIAL SIGNAL CLASSIFICATION

Raymond Brueckner[1,2], *Björn Schuller*[3,1,4]

[1]Machine Intelligence & Signal Processing Group, MMK,
Technische Universität München, Germany
[2]Nuance Communications Deutschland GmbH, Aachen, Germany
[3]Department of Computing, Imperial College London, UK
[4]Institute for Sensor Systems, University of Passau, Germany

raymond.brueckner@web.de, bjoern.schuller@imperial.ac.uk

ABSTRACT

With the impressive advances of deep learning in recent years the interest in neural networks has resurged in the fields of automatic speech recognition and emotion recognition.

In this paper we apply neural networks to address speaker-independent detection and classification of laughter and filler vocalizations in speech. We first explore modeling class posteriors with standard neural networks and deep stacked autoencoders. Then, we adopt a hierarchical neural architecture to compute enhanced class posteriors and demonstrate that this approach introduces significant and consistent improvements on the Social Signals Sub-Challenge of the Interspeech 2013 Computational Paralinguistics Challenge (ComParE). On this task we achieve a value of 92.4% of the unweighted average area-under-the-curve, which is the official competition measure, on the test set. This constitutes an improvement of 9.1% over the baseline and is the best result obtained so far on this task.

***Index Terms*—** enhanced posteriors, hierarchical neural networks, deep autoencoder networks, computational paralinguistics challenge

1. INTRODUCTION

The emerging field of computational paralinguistics is dedicated to the study of non-verbal elements of speech that convey information about human affect, emotion, personality, and speaker states and traits. There is an increasing amount of research in that field [1][2][3][4] and a number of Interspeech challenges in recent years have been organized with the intention to foster research in the many different aspects of paralanguage and to combine the sometimes scattered research efforts leveraging synergy effects [5].

In this paper we introduce hierarchies of neural networks and explore their effect on the classification performance on the Sub-Challenge task. We show how adopting these networks naturally leads to a smoothed and enhanced variant of the posterior probabilities commonly obtained at the output of standard multi-layer perceptrons (MLP). The time trajectories of these enhanced posterior probabilities lead to better classification performance and generalize well. Next, we examine if replacing these standard MLP with deep networks, such as stacked autoencoders (SAE), improves the results. In previous work [6] we showed for the Likability Sub-Challenge classification task of the Interspeech 2012 Speaker Trait Challenge [7] that the modeling power of Deep Belief Networks (DBN) could not be leveraged. This was most probably due to the severe overfitting that occurred as the relevant task was based on utterance-wise feature vectors. In the Social Signals Sub-Challenge, however, frame-based acoustic features are used. Therefore, overfitting does not pose any problem. We evaluate different network architectures employing varying ranges of feature-level context. Further, we explore the effect of different number of hidden units in the MLP and SAE and the number of hidden layers in the SAE.

We explain the concept of enhanced posteriors in Section 2, before giving a brief outline of autoencoder networks in Section 3. The experimental results are detailed in Section 4.

2. ENHANCED POSTERIORS

The use of posterior probabilities has become popular for improving automatic speech recognition (ASR) systems and has been extensively studied in the past [8][9][10]. There exist two general ways to adopt posteriors: In the hybrid Hidden Markov Model / Artificial Neural Network (HMM/ANN) approach [11] the posterior probabilities are used as local acoustic scores, while in the Tandem approach [12] the posterior probabilities are fed as acoustic features into a HMM system, usually after applying some transformation (e. g., PCA, LDA, or logarithm) on the features.

In both cases Multi-Layer Perceptrons (MLP) have tradi-

978-1-4799-2756-2/13/$31.00 ©2013 IEEE

tionally been used to estimate the posteriors. In recent years this idea has been extended to using deep networks of various architectures and has led to a significant performance boost on a wide range of tasks [13][14][15]. Instead of estimating the posteriors with a single-hidden layer neural network, two or more hidden layers are used. In the feed-forward evaluation phase this may still be called a MLP, but different names have been coined in the literature, e. g., Deep Belief Network (DBN) [16], Stacked Autoencoder (SAE) [17], etc., depending on how the deep network has been pre-trained.

Another technique to improve upon the performance of posterior-based systems is to build a second network on top of the first one, thus building a *hierarchical* neural network. This idea has previously been described for ASR systems [18] and was shown to improve results. In this paper we will show that this idea can successfully be employed also in the field of social signal classification. Instead of optimizing the network on a phone alignment we will optimize our networks on the given target class labels.

In the following we will refer to the first layer posteriors as *regular* or *first-order* posteriors and to any higher-layer posteriors as *enhanced* or *higher-order* posteriors.

In order to model temporal context within neural networks a common approach is to stack a fixed number of n successive frames, so that a sequence of feature vectors is presented to the network at each time step [19]. Often an equal number of past and future feature frames around the central feature vector x_t is agglomerated. A sliding window from $t-(n-1)/2$ to $t+(n-1)/2$ is applied to merge n successive feature vectors of size N to an $n \cdot N$-dimensional extended feature vector x'_t, i. e.,

$$\begin{aligned} x'_t &= [x_{t-\frac{n-1}{2}}; ...; x_t; ...; x_{t+\frac{n-1}{2}}] \\ \text{for} \quad & \frac{n-1}{2} < t \leq T - \frac{n-1}{2} \, . \end{aligned} \tag{1}$$

In order to obtain valid vectors for $t \leq (n-1)/2$ and $t > T-(n-1)/2$, the first and last feature vector of $x_{1:T}$ needs to be padded $(n-1)/2$ times.

The extended feature vector x'_t is then fed into the first MLP as input. The trained network transforms the input features into regular posteriors. These can be stacked into an extended posterior vector just the same way as explained above. This vector serves as input to a second MLP, which can be learned based on the regular posteriors in order to learn long-term inter- and intra-dependencies between class evidences (posteriors) in the training data and transform the regular posteriors into enhanced posteriors. Figure 1 shows a schematic example of a network transforming a temporal context of n stacked input frames into a vector of enhanced posteriors.

The first MLP receives the stacked baseline (acoustic) features as input and estimates class posterior probabilities on its output nodes. Subsequently, the second MLP uses a long context of regular class posteriors as input and estimates enhanced class posteriors on its output. Here, we used the same database for training the two MLPs. The long term dependencies captured by the higher MLP leads to an enhancement of the quality of the class posteriors. The rational behind this is that at the output of every MLP, the information stream gets simpler (converging to a sequence of binary posterior vectors), and can thus be further processed (using a simpler classifier) by looking at a larger temporal window [18].

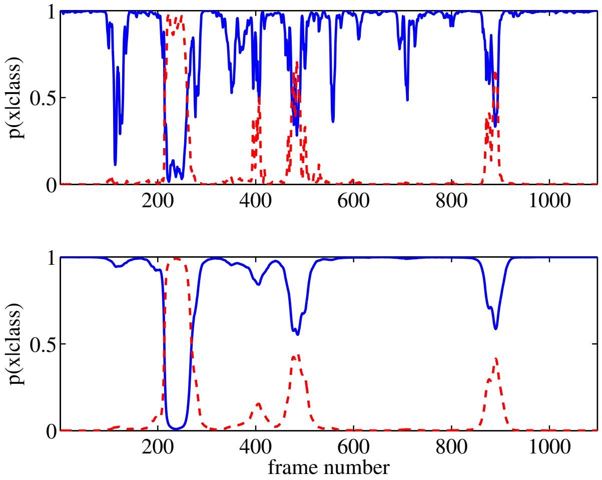

Fig. 2. *Example of a posteriorgram showing the posterior trajectories over time for one utterance. The plot on the top shows the posteriorgram of the regular posteriors for the two classes* garbage *(solid blue line) and* laughter *(dotted red line). The plot on the bottom shows the posteriorgram of the enhanced posteriors for the same classes and utterance.*

A plot of the values of the posteriors over time is referred to as a *posteriorgram* [20]. A typical example of a posteriogram for the Social Signals database is given in Figure 2.

What is evident from the plot is that the enhanced posteriors are much smoother than their regular counterparts. They also exhibit less spiky behavior which usually leads to more false alarms; this has often been tackled by some form of heuristic smoothing [21]. A downside of this smoothing are the shallower ramps at the class boundaries. We conjecture that there will be more errors in these transition areas.

3. AUTOENCODER NETWORKS

An autoencoder (AE) is an artificial neural network that tries to learn a compressed representation for its input data. This is accomplished in the following way: given a set of input feature frames an AE computes the hidden layer activations, usually adopting a non-linear activation function, such as the sigmoid function. This is referred to as the (*encoding* phase). It then tries to reconstruct the input by computing the output activations given the hidden layer activations (*decoding* phase)

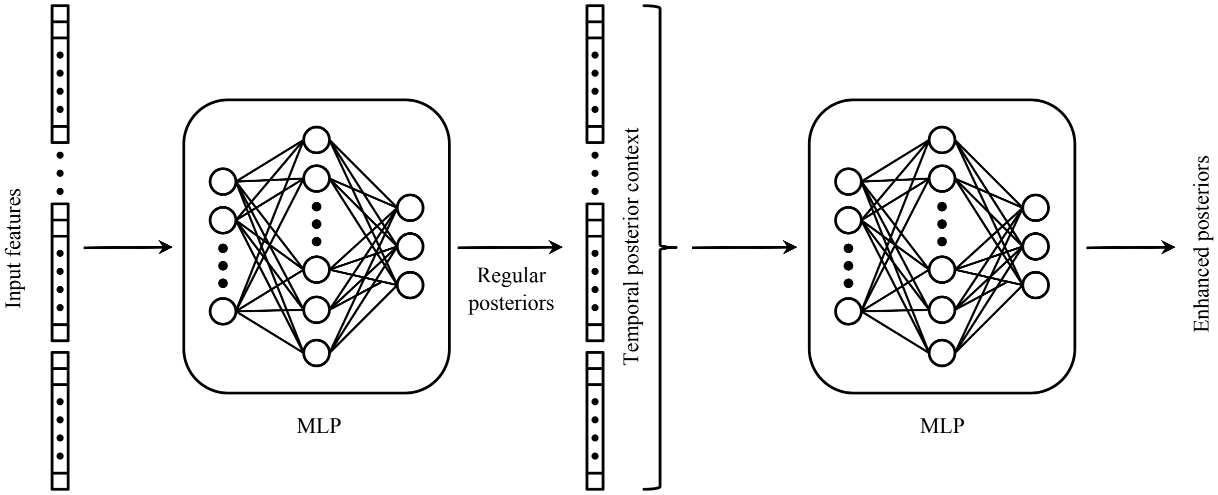

Fig. 1. *Hierarchical network to generate enhanced posteriors: The first MLP transforms stacked (acoustic) features into regular posteriors. A temporal context of those posterior vectors is created by frame stacking. The second MLP processes the temporal context of regular posteriors and learns long term dependencies to estimate enhanced posteriors.*

with the target being identical to the input. In the output layer one usually adopts a non-linear function for binary input and a linear function for real-valued input. The cost function to be minimized generally is chosen to be the mean-squared error (MSE) for real-valued input/output or the cross-entropy for binary input/output.

It should be noted that without any further constraints successfully training an autoencoder network requires the hidden layer to be smaller than the input layer. Otherwise the encoding will easily learn the identity function, which is the trivial solution the minimization problem. This approach is generally referred to as the *bottleneck* architecture. However, a number of alternative architectures have been proposed to avoid this constraint, such as the *denoising* autoencoder [22] or the *contractive* autoencoder [23].

The main motivation for adopting autoencoder networks is to pre-train - possibly deep - neural networks in an unsupervised manner. This pre-training moves the network parameters close to an optimum and thus gives a good initialization to a subsequent fine-tuning step, e. g., by running Stochastic Gradient Descent (SGD).

Moreover, it is possible to stack the resulting, pre-trained autoencoders to form a deep *stacked autoencoder* to get a good initialization for a deep network, which can subsequently be fine-tuned. An alternative approach is to use Restricted Boltzmann Machines (RBM), which has been investigated earlier on the task of Likability Classification [6]. As a pre-training step for deep networks it is debatable whether RBMs or AEs give better performance. In practice they seem to give comparable results on many tasks. Some informal experiments we have conducted on the current Sub-Challenge has confirmed this and as AEs are somewhat faster to train, we have decided to prefer AEs over RBMs.

4. EXPERIMENTS

4.1. Database and feature set

The results presented in this section were obtained by running experiments on the Social Signals Sub-Challenge of the Interspeech 2013 Computational Paralinguistics Challenge (ComParE), which comprises 2763 utterances or roughly 3 million frames in total. The task is to perform a frame-wise classification of three vocalization classes during phone conversations between two persons, where the voice of only one speaker is audible. The classes are: *laughter*, *filler* (vocalizations such as "uhm", "eh", "ah", etc.), and *garbage*, which contains all other vocalizations, such as speech, further also including silence. The results reported in this paper are based on the baseline feature set composed of 141 features. For details about the Challenge and the underlying baseline feature set refer to [24].

4.2. Regular posteriors

For the experiments on regular posteriors we trained all networks on the frame-wise class targets of the full training set. As network input x_t we used the full competition baseline feature set comprising all 141 features. For feature frame stacking, we evaluated sliding windows of lengths between $n = 1$ and $n = 15$. Given the frame shift of 10 ms and a frame size of 20 ms this amounts to a maximum temporal context of approximately 160 ms, which is in the range of average phone durations of human speech [25].

For training the networks we used standard Stochastic Gradient Descent (SGD) using momentum. Further, we applied L_2-regularization on the layer weights. All meta-parameters used to train the networks such as the number and size of the hidden layers, learning rate , momentum, and batch size were chosen to be the ones that gave the highest unweighted average area-under-the-curve (UAAUC) value on the development set.

We evaluated two different network setups: single-layer MLP without pre-training and multi-layer MLPs with stacked autoencoder (SAE) pre-training. Contrary to the results reported in [6], informal experiments on the Social Signals database have shown that pre-training a single-layer MLP does not improve performance.

Table 1 compares the UAAUC for a single-hidden layer MLP and a two-hidden layer SAE for different layer sizes.

UAAUC [%]	size of hidden layer(s)				
	64	128	256	512	1024
MLP	92.5	92.8	**93.0**	92.8	92.7
Deep SAE (2)	93.1	93.4	**93.7**	93.4	93.3

Table 1. *Regular posteriors: Comparison of a single-hidden layer MLP and a two-hidden layer SAE for different hidden layer sizes on the development set.*

Based on these findings we fixed the layer size to 256 and investigated how the number of layers in a deep SAE would affect the performance. Table 2 shows the results.

UAAUC [%]	number of hidden layers				
	1	2	3	4	5
Deep SAE	93.0	**93.7**	93.4	93.2	93.0

Table 2. *Regular posteriors: Effect of the number of hidden layers in a deep SAE on the UAAUC on the development set.*

Best results were obtained with 2 hidden layers only. We conjecture that this is due to the relatively few (only three) classes, so that no advantage can be drawn from the presumed higher modeling power of deeper nets. However, this requires a more thorough analysis.

On top of the experiments described above, we also tried different temporal context sizes (results not shown here), but a context of 11 frames gave the best results.

4.3. Enhanded posteriors

For training the enhanced or *second-order* posterior networks we followed the approach described in Section 4.2: We took the three-dimensional regular posterior vectors and applied sliding windows of lengths between $n = 3$ and $n = 201$ for stacking the frames, which amounts to a maximum temporal context of approximately 200 ms.

The set of meta-parameters to be optimized was the same as the one used for the regular posteriors. Again we chose the ones that gave the highest UAAUC value on the development set.

First, we investigated the effect of different context lengths of regular posteriors which were used as input to the second network generating the enhanced posteriors. Table 3 shows the results for a MLP with a hidden layer of 256 units.

UAAUC [%]	# context frames					
	51	75	101	151	175	201
MLP	96.6	96.9	97.1	**97.3**	97.2	97.1

Table 3. *Enhanced posteriors: Comparison of the effect of the temporal context of stacked regular posteriors for a MLP with 256 hidden units on the development set.*

We obtained best results for a context size of 151 frames. With this value we achieved an UAAUC of 97.3%. This is an impressive improvement of 9.7% absolute over the baseline on the development set. The table further shows that the performance is not overly sensitive to the context size.

Next, using this setup, we varied the number of hidden units in the network. The results are depicted in Table 4.

UAAUC [%]	# hidden units					
	64	128	256	512	1024	2048
MLP	96.8	97.1	**97.3**	97.2	97.2	97.1

Table 4. *Enhanced posteriors: Comparison of the effect of the number of hidden units for a MLP using an input context of 151 frames on the development set.*

The table confirms the previously chosen value of 256 as the optimal hidden layer size for the enhanced posterior network. Again, we observe that the decrease in performance is rather small as we move away from the optimum number of hidden units.

Due to limitations in the available training time we were unable to investigate deep SAEs on the regular posteriors to generate the enhanced posteriors. We plan to investigate this issue in the future.

4.4. Higher-order enhanced posteriors

In the spirit of generating enhanced posteriors built from the regular posteriors we have also tried to stack another MLP on top of the current system and use the (second-order) enhanced posteriors as input to generate higher-order enhanced posteriors. Just as described in Section 4.2 we have taken a context of enhanced posterior frames and used the stacked frames as input to yet another MLP. The outputs of this trained network still represent posteriors - we refer to them as *third-order* posteriors. The results of using these higher-order posteriors are given in Table 5.

Comparing these results with those shown in Table 3 we observe that for shorter, sub-optimal context lengths (51 is

# context frames (regular)	51		151	
# context frames (enhanced)	51	151	51	151
UAAUC [%]	96.8	96.9	**97.1**	96.8

Table 5. *Third-order posteriors: Results obtained for a 2nd-order MLP on the development set. The first row shows the number of frames of regular posteriors (output from the first MLP) used to build the input of the second MLP. The second row shows the number of frames of enhanced posteriors (output from the second MLP) used to build the input to the third MLP.*

this case) higher-order posteriors give rise to a slight improvement. However, for the optimum context length of 151 frames the performance slightly decreases. We suspect that this is due to the effect of overly smoothing the posterior trajectories, especially at the transition boundaries between classes.

In summary, for the task at hand going beyond second-order posteriors does not further redound to performance improvements.

4.5. Summary

In the following we summarize the best results obtained on the Sub-Challenge. Note that we have strictly adhered to the challenge rules which in particular imposed a maximum of 5 submissions of results obtained on the test data.

In Table 6 we show the baseline results together with the results of our best setups for regular posteriors and for enhanced, i. e., second-order, posteriors. We report the AUC and UAAUC measures obtained on the development set, which served as the basis for choosing the optimal parameters as well as the numbers for the test set.

[%]		devel set	test set
baseline	AUC [Laughter]	86.2	82.9
	AUC [Filler]	89.0	83.6
	UAAUC	87.6	**83.3**
regular posteriors	AUC [Laughter]	92.8	90.5
	AUC [Filler]	94.5	88.0
	UAAUC	93.7	**89.2**
enhanced posteriors	AUC [Laughter]	98.1	94.9
	AUC [Filler]	96.5	89.9
	UAAUC	97.3	**92.4**

Table 6. *Summary of best results. Depicted are results on the development and the test set using models trained on the full training set. Only the test results for the baseline were obtained training on the training and development set.*

Note that for the results of the baseline on the test set the respective models were retrained on the union of the training and development sub-sets. On the contrary, retraining our networks on both sub-sets, the results slightly worsened, so our results on the test set are based on networks that were trained on the training set only.

5. CONCLUSIONS

We have successfully applied a hierarchical neural network architecture that generates enhanced posterior probabilities on the problem of classifying the three different classes *garbage*, *laughter*, and *filler* of the Social Signals Sub-Challenge of the Interspeech 2013 Computational Paralinguistics Challenge. Exploiting temporal contextual information over the regular class posteriors the enhanced posteriors exhibit smoothed time trajectories yielding substantial improvements over the regular posteriors.

In adopting our approach we view the task as a conventional classification task and manage to obtain a UAAUC of 92.4% on the test set, an increase of 9.1% absolute over the baseline result. This is the best result on this task reported so far in the literature, outperforming the Sub-Challenge winner's results [26] from the Interspeech 2013, while strictly adhering to the challenge rules.

A promising direction for future research is, hence, exploring upsampling or downsampling the data of the respective classes. Further, instead of treating the problem as a pure classification task, approaching it using keyword or detection techniques and a combination of these with the presented strategy might yield further improvements. We also plan to feed the enhanced posterior features into sequential models, such as HMMs or recurrent neural networks in order to exploit their temporal modeling capacities.

6. ACKNOWLEDGEMENT

The research presented in this publication was conducted while the first author was employed by Nuance Communications Deutschland GmbH.

7. REFERENCES

[1] B. Schuller, "The Computational Paralinguistics Challenge," *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 97–101, July 2012.

[2] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, F. Burkhardt, and R. van Son, "Introduction to the Special Issue on Next Generation Computational Paralinguistics," *Computer Speech and Language, Special Issue on Next Generation Computational Paralinguistics*, 2014, to appear.

[3] B. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*, Wiley, 2013, to appear.

[4] Z. Zhang, J. Deng, and B. Schuller, "Co-Training Succeeds in Computational Paralinguistics," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 8505–8509.

[5] B. Schuller and F. Weninger, "Ten Recent Trends in Computational Paralinguistics," in *4th COST 2102 International Training School on Cognitive Behavioural Systems*, vol. 7403/2012, pp. 35–49. Springer, 2012.

[6] R. Brueckner and B. Schuller, "Likability Classification - A Not so Deep Neural Network Approach," in *Proc. of Interspeech*, Florence, Italy, 2012.

[7] B. Schuller, S. Steidl, A. Batliner, E. Nöth, A. Vinciarelli, A. Burkhardt, R. van Son, F. Weninger, F. Eyben, T. Bocklet, G. Mohammadi, and B. Weiss, "The Interspeech 2012 Speaker Trait Challenge," in *Proc. of Interspeech*, Portland, OR, USA, 2012.

[8] S. Thomas, P. Nguyen, G. Zweig, and H. Hermansky, "MLP based phoneme detectors for Automatic Speech Recognition.," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 5024–5027.

[9] S. Soldo, M. Magimai-Doss, J. Pinto, and H. Bourlard, "Posterior features for template-based ASR.," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 4864–4867.

[10] P. Fousek and H. Hermansky, "Towards ASR Based On Hierarchical Posterior-Based Keyword Recognition.," in *Proc. of ICASSP*, Toulouse, France, 2006, pp. 433–436.

[11] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[12] H. Hermansky, D. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," in *Proc. of ICASSP*, Istanbul, Turkey, 2000, pp. 3476–3479.

[13] M. Abdel-Rahman, G. Dahl, and G. E. Hinton, "Acoustic Modeling using Deep Belief Networks.," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[14] F. Seide, G. Li, X. Chen, and D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," in *Proc. of ASRU*, Hawaii, USA, Dec 2011, pp. 24–29.

[15] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep Neural Networks for Acoustic Emotion Recognition: Raising the Benchmarks," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, pp. 5688–5691.

[16] I. Sutskever and G. E. Hinton, "Deep, Narrow Sigmoid Belief Networks Are Universal Approximators.," *Neural Computation*, , no. 11, pp. 2629–2636, 2008.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[18] H. Ketabdar and H. Bourlard, "Enhanced phone posteriors for improving speech recognition systems," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1094–1106, 2010.

[19] M. Wöllmer, B. Schuller, and G. Rigoll, "Feature Frame Stacking in RNN-Based Tandem ASR Systems - Learned vs. Predefined Context," in *Proc. of Interspeech*, Florence, Italy, 2011, pp. 1233–1236.

[20] M. J. R. Gomez and D. P. W. Ellis, "Error visualization for tandem acoustic modeling on the aurora task.," in *Proc. of ICASSP*, Orlando, FL, USA, 2002, pp. 4176–4179.

[21] Y. Sun, D. Willett, R. Brueckner, R. Gruhn, and D. Bühler, "Experiments on Chinese speech recognition with tonal models and pitch estimation using the Mandarin speecon data," in *Proc. of Interspeech*, Pittsburgh, PA, USA, 2006, pp. 1245–1248.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of ICML*, New York, NY, USA, 2008, pp. 1096–1103.

[23] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. of ICML*, Bellevue, WA, USA, 2011, pp. 833–840.

[24] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, F. Chetouani, M. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The Interspeech 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism," in *Proc. of Interspeech*, Lyon, France, 2013, pp. 148–152.

[25] B. Zilko and M. Zilko, "Time Durations of Phonemes in Polish Language for Speech and Speaker Recognition.," in *LTC*. 2009, Lecture Notes in Computer Science, pp. 105–114, Springer.

[26] R. Gupta, K. Audhkhasi, S. Lee, and S. Narayana, "Paralinguistic event detection from speech using probabilistic time-series smoothing and masking," in *Proc. of Interspeech*, Lyon, France, 2013, pp. 173–177.