INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

DBN-ivector Framework for Acoustic Emotion Recognition

Rui Xia, Yang Liu

Computer Science Department, The University of Texas at Dallas, Richardson, TX 75080, USA
rx,yangl@hlt.utdallas.edu

Abstract

Deep learning and i-vectors have been successfully used in speech and speaker recognition recently. In this work we propose a framework based on deep belief network (DBN) and i-vector space modeling for acoustic emotion recognition. We use two types of labels for frame level DBN training. The first one is the vector of posterior probabilities calculated from the GMM universal background model (UBM). The second one is the predicted label based on the GMMs. The DBN is trained to minimize errors for both types. After DBN training, we use the vector of posterior probabilities estimated by DBN to replace the UBM for i-vector extraction. Finally the extracted i-vectors are used in backend classifiers for emotion recognition. Our experiments on the USC IEMOCAP data show the effectiveness of our proposed DBN-ivector framework. In particular, with decision level combination, our proposed system yields significant improvement on both unweighted and weighted accuracy.
Index Terms: Emotion recognition, DBN, i-vector

1. Introduction

There has been a lot of research efforts on acoustic emotion recognition in recent years. Because human's emotion is quite complex in terms of both generation and perception, how to predict accurate affective information from natural and spontaneous speech is still a challenging task. Most of past research on this problem has focused on extracting discriminative features and developing robust models. One of the most successful systems is based on supra-segmental acoustic features and has shown great performance in many challenges recently [1, 2]. In this method, various statistical functions are applied to different types of low level acoustic features. The derived high dimensional features can be used in traditional back-end classifiers. Another type of systems uses frame-level dynamic features (e.g., MFCC features). Different methods have been proposed to model these features, for example, Gaussian Mixture Model (GMM), HMM, and i-vector space modeling. Among them, the i-vector approach has shown competitive system performance recently on affective computing related tasks [3, 4, 5, 6]. The i-vector extraction process aims to obtain the most important variations from high-dimensional supervectors based on GMMs. Then these i-vectors can be used as features in back-end classifiers (see Sec. 2 for more descriptions).

In the past few years, deep learning has become the standard approach for speech recognition. The role of all types of deep networks, such as deep neural networks (DNN) or convolution neural networks (CNN), is often to replace the traditional GMMs. These deep networks have the ability to learn high-level hidden features [7]. They are discriminatively trained, whereas the standard GMM is a generative model. In [8] and [9], the authors investigated training DNN and CNN to replace GMMs in speaker identification and language identification. The DNN and CNN are trained to minimize the labeling error for the tied-triphones. The outputs from the deep networks are used to calculate the sufficient statistics for extracting i-vectors. Improved speaker and language identification performance is observed using such i-vectors.

Deep learning techniques have also been applied to the acoustic emotion recognition task [10] [11]. In this paper, we propose to use DBN in the paradigm of i-vector space modeling approach for acoustic emotion recognition in order to combine the advantages of the two approaches. The key idea is to use a discriminately trained DBN to estimate the posterior probabilities of the mixture components for each frame and use these for i-vector extraction. Our proposed method has the following steps. The first step generates two kinds of labels for each frame. One is the posterior probabilities of the mixture components of the universal background model (UBM) that is trained using all the data. The other one is the predicted emotion label based on the likelihood of the emotional GMMs that are trained with the corresponding training utterances. In the second step, a DBN is built and trained to simultaneously minimize two kinds of loss functions with respect to the two different labels above: one loss is the cross entropy between the DBN's outputs and the frame's posterior probabilities based on the UBM; and the other is the negative log-likelihood of the predicted label. For DBN training, we evaluate two input variations of the self labels from the UBM and GMMs: i) for each frame, we only use the top N posterior probabilities and zero out the others in the vector of posterior probabilities; ii) the vector of posterior probabilities and the log likelihood from the emotional GMM are smoothed by taking the average of the labels for the current frame with those from the previous and the following frames. In the last step, after training DBN, its output is used to replace the posterior probability of the UBM to calculate the zero order and first order statistics, and then i-vector is extracted and fed into the back-end classifier for emotion recognition. Our experimental results on the USC IEMOCAP data show our proposed method outperforms the standard i-vector or DBN methods for emotion recognition.

2. i-vector space modeling and Deep Belief Network framework

Both i-vector space modeling and DBN are used as a transformation method to change the features into another representation, which is further used in a back-end classifier for emotion recognition. In the following we briefly describe the i-vector extraction and DBN training methods.

2.1. i-vector space modeling

I-vector space modeling approach maps the high dimensional GMM supervector space (generated from concatenating all the

http://dx.doi.org/10.21437/Interspeech.2016-488

mean values of GMM) to low dimensional total variability space which contains most variation between segments. The target GMM supervector can be viewed as shifted from the UBM. Formally, a target GMM supervector M can be written as:

$$M = m + Tw \tag{1}$$

where m represents the UBM supervector, T is a low dimensional rectangular total variability matrix, and w is termed as i-vector. In order to estimate total variability matrix T and i-vector w^u for a given utterance u, Baum-Welch statistics, zero order N^u and first order F^u, are needed and can be calculated as the following:

$$N_c^u = \sum_{t=1}^{L} P(c|x_u^t) \tag{2}$$

$$F_c^u = \sum_{t=1}^{L} P(c|x_u^t) x_u^t \tag{3}$$

where c represents the index of Gaussian mixture component in GMM, x_u^t is the t-th frame in utterance u, and $P(c|x_u^t)$ is the posterior probability of the c^{th} mixture. Given F_c^u and GMM, the centralized first order statistics can be obtained as follows:

$$\overline{F_c^u} = \sum_{t=1}^{L} P(c|x_u^t)(x_u^t - m_c) \tag{4}$$

where m_c is the c-th mean of the given GMM. With sufficient statistics, total variability matrix T is estimated through EM algorithm introduced in [12] and latent vector w^u can be extracted. [13] describes more details about how to extract i-vectors.

2.2. Deep Belief Network

DBN is constructed by stacking more than one Restricted Boltzmann Machines (RBMs), which is one special case of undirected graphical models [7]. This framework learns to extract meaningful hidden hierarchical representation from the training data. The training process of DBN is divided into two steps, unsupervised pre-training and supervised fine-tuning.

The pre-training stage typically is done in a greedy layer-wised manner. In this work, Gaussian-RBM is used as the layer component in order to model the real-valued acoustic inputs. Given input data v as the visible nodes and hidden variable h as the hidden nodes, the joint-probability of Gaussian-RBM is:

$$p(v) = \frac{\sum_h e^{-E(v,h)}}{Z} \tag{5}$$

where Z is the partition function, and the energy function $E(v,h)$ is calculated as follows:

$$E(v,h) = \sum_{i \in vis} \frac{1}{2\sigma_i^2}(v_i - b_i)^2 - \sum_{j \in hid} c_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \tag{6}$$

where w_{ij} is the connection matrix between the hidden and visible nodes, σ_i is the standard deviation of the visible unit i, and b and c are the bias vectors for the visible and hidden nodes respectively. The learning process is to minimize the empirical negative log-likelihood of the training data. The approximate algorithm called Contrastive Divergence (CD) is applied to update parameters efficiently. In this DBN framework, we use the Noisy Rectified Linear Unit (NReLU) [14] as the non-linear activation function instead of using the sigmoid function. We also add an upper-bound to avoid having hidden nodes with large values. The upper-bound value in this paper is equal to 1. More information about NReLU used in DBN can be found in [14, 15, 16].

After pre-training, the parameters of DBN are used as the initial values and further tuned in the subsequent supervised fine-tuning stage. The detail of the fine-tuning stage used in our method is described in the following section.

3. DBN-ivector framework

In this section, we will describe our proposed DBN-ivector framework for emotion recognition. Deep networks have been widely adopted in ASR systems for acoustic modeling. For deep network training in ASR, it is straightforward to use labels such as phones or tri-phones. However, for acoustic emotion recognition, it is not easy to obtain accurate frame level labels. This is because on one hand it is hard for human annotators to assign emotion labels to frames, and on the other hand the automatic classifiers are not accurate, unlike forced alignment in ASR training, resulting in unreliable predicted labels. In this work, we use two different frame labels for discriminate DBN training. The details of our proposed method are as follows.

There are three major components in our framework, as shown in Figure 1. In the first component (the left box in Figure 1), we generate two types of labels for each frame.

- Posterior probabilities of mixture components of the UBM: We train the UBM of MFCC features using all the training data. For an utterance u, x_t^u represents the feature set of t-th frame. Given x_t^u, a vector of posterior probabilities, $\boldsymbol{P}(c|x_t^u, \Omega) = [p(c_1|x_t^u, \Omega), p(c_2|x_t^u, \Omega), ..., p(c_M|x_t^u, \Omega)]$ can be calculated, where Ω is the UBM and M is its number of mixtures.
- Predicted emotional label by GMMs: For L emotion classes, we train L emotional GMMs with the corresponding utterances for each emotion. Then the log likelihood, $LLK(x_t^u|\Omega_j)$, can be calculated for each frame using each GMM Ω_j, and the prediction, $pred(x_t^u)$, can be obtained by choosing the largest $LLK(x_t^u|\Omega_j)$.

The second component is DBN training, shown in the middle in Figure 1. In our work, the pre-training step is the same as the traditional DBN introduced in the previous section. However, the fine-tuning stage is different since here we consider two optimization targets, corresponding to the two kinds of labels described above.

- For the predicted emotion label of x_t^u, we minimize the negative of the log likelihood of the given prediction, $Pred(x_t^u)$. The softmax layer is put on top of the last hidden layer h_l for classification. The loss function is as follow:

$$p(pred(x_t^u)|h_l, x_t^u) = \frac{e^{h_l^T W_k + B_k}}{\sum_j e^{h_l^T W_j + B_j}} \tag{7}$$

  where W and B are the parameters of the soft-max layer.
- For the posterior probabilities of the mixture components, we use cross entropy [17] . One additional hidden layer h_d is put on top of the last hidden layer. The number of hidden nodes is equal to the size of $\boldsymbol{P}(c|x_t^u)$, i.e.,

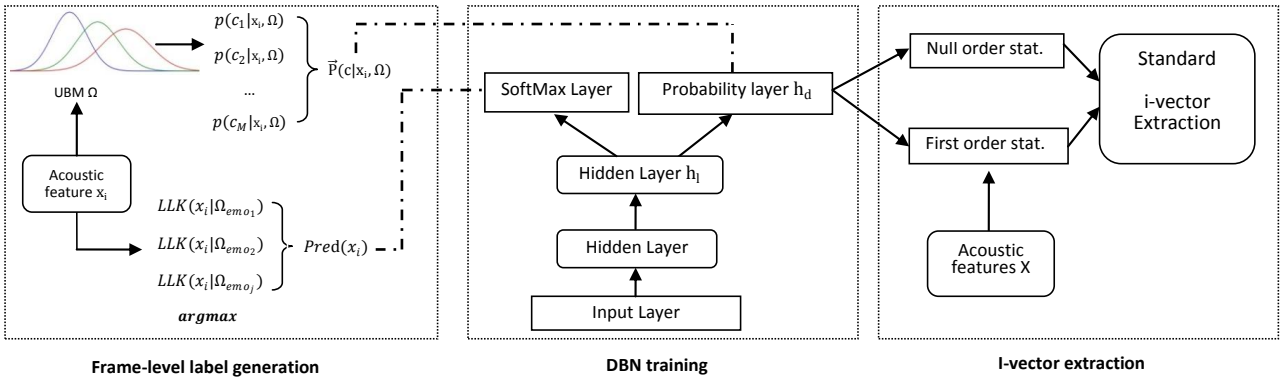

Figure 1: DBN-ivector framework

the number of mixture components of UBM. Since the posterior probabilities are in the range of $[0, 1]$, we use a sigmoid function, $h_d = sigmoid(W_d * h_l + b_d)$, where W_d and b_d are the weight and bias matrix connected with h_l. The optimization process is to reduce the cross entropy error between h_d and $\boldsymbol{P}(c|x_t^u)$. The cross entropy error is calculated as follows:

$$Loss_{ce}(h_d, \boldsymbol{P}(c|x_t^u)) = -[h_d log(\boldsymbol{P}(c|x_t^u)) + (1 - h_d) log(1 - \boldsymbol{P}(c|x_t^u))] \quad (8)$$

Considering both labels, the loss function used in the DBN is thus:

$$Loss = Loss_{ce}(h_d, \boldsymbol{P}(c|x_t^u)) + \alpha * (-p(pred(x_t^u)|h_l)) \quad (9)$$

where α is the hyper parameter for balancing the two types of losses (we use 0.5 in this work). With iterative training, the DBN is trained to simultaneously lower the combined cross entropy and classification error.

Because there is noise in the two kinds of self-generated labels, we propose two approaches to vary the labels for DBN training. These are also motivated by the observation that human emotions are highly related to long temporal information and frame-level labeling is not accurate. The followings are the two methods we evaluate.

- Instead of using posterior probabilities for all the mixture components, we use a selection strategy by keeping the top N posterior probabilities in the vector $\boldsymbol{P}(c|x_t^u)$, and setting the others to zero. This is expected to keep only the most confident mixture components and remove the noise from other unreliable ones.
- We use a smoothing strategy for the posterior probabilities and the likelihood scores by taking the average of the values from a window including the current frame and its previous and following K frames:

$$\overline{\boldsymbol{P}(c|x_t^u)} = \frac{1}{2k+1} \sum_{i=t-k}^{t+k} \boldsymbol{P}(c|x_i^u) \quad (10)$$

$$\overline{LLK(x_t^u|\Omega_j)} = \frac{1}{2k+1} \sum_{i=t-k}^{t+k} LLK(x_t^u|\Omega_j) \quad (11)$$

After smoothing, $\overline{\boldsymbol{P}(c|x_t^u)}$ is used as the vector of posterior probabilities for x_t^u. The prediction of x_t^u, $Pred(x_t^u)$, can be obtained by taking the $argmax$ function on $\overline{LLK(x_t^u|\Omega_j)}$. Such smoothing allows us to use longer range information than a single frame, which is consistent with human's perception of emotion.

After training DBN, h_d is calculated by feedforwarding through the network for every frame and taken as the input of the third step (right box in Figure 1). Each output in h_d, corresponding to one mixture component, replaces $p(c|x)$ in Equation 2 and 3. With sufficient statistics, i-vector w can be obtained from the standard i-vector extraction process, and then fed into SVM for training the emotion recognition model.

Note that Fig 1 shows the training process of our DBN-ivector framework. For testing, we feed the MFCC features of each frame to the DBN model, and use the posterior probability output from the DBN and the GMM supervectors to calculate the i-vector and then obtain the SVM's emotion prediction.

4. Experiment

4.1. Data

In this work, we use Interactive Emotional Dyadic Motion Capture (IEMOCAP) [18] to evaluate our proposed method. This corpus has approximately 12 hours of audiovisual data, including video, speech, motion capture of face, and text transcriptions [18]. It has 10 professional actors (5 male and 5 female) acting in two different scenarios: scripted play and spontaneous dialog, in their dyadic interactions. Each interaction is around 5 minutes in length, and is segmented into sentence levels. We use four emotion categories in this study: angry, happy, sad, and neutral, similar to most prior studies using this corpus. Note that we merged 'happy' and 'excited' in the original annotation into the 'happy' class. Only the utterances with the majority agreement are used in the experiments. In total we use 5,531 utterances. The class distribution is: 20.0% angry, 19.6% sad, 29.6% happy, and 30.8% neutral.

4.2. Experiment setup

The experiment protocol for IEMOCAP data is leave-one-speaker-out which means there is no speaker overlap between training and testing set. System performance is evaluated by two metrics, weighted accuracy (WA) and unweighted accuracy

Table 1: Emotion recognition result (%) for IEMOCAP

Systems		WA	UA
Standard i-vector		56.4	57.5
DBN-ivector		56.7	58.1
DBN-ivector top N	N = 2	**57.2**	**58.6**
	N = 4	56.6	57.9
	N = 8	56.8	58.1
DBN-ivector smooth K window	$K = 2$	**57.1**	**58.4**
	$K = 4$	57.0	58.3
	$K = 7$	56.6	58.2
	$K = 10$	56.4	57.8

(UA). Both metrics are standard measurements used in several previous emotion challenges.

The acoustic features we use are 39 dimensional MFCC including first and second derivatives. The number of mixtures of the UBM is 16 in this work. For DBN, the input is also 39 dimensional MFCC. Three hidden layers are used in DBN and each of them has 100 hidden nodes. In the pre-training stage, the learning rate is 0.0001 and the number of training iterations is 5. Note that we use only the training data for pre-training. In the fine-tuning stage, we use the ADADELTA algorithm for optimizing the loss function because it does not need to preset the global learning rate and it has a much faster converging rate [19]. The fine-tuning iteration number is set to 5. We extract 100 dimension i-vector through the standard i-vector extraction framework. Once the i-vector is obtained, we use a linear kernel SVM with sequential minimal optimization (SMO) as the back-end classifier for emotion recognition.

4.3. Result

The experimental results are show in Table 1. The baseline system is the standard i-vector system that uses the sufficient statistics directly calculated from the UBM. The WA and UA of the baseline system is 56.4% and 57.5% respectively. Our proposed DBN-ivector system without the top-N selection or smoothing for DBN training achieves 56.7% for WA and 58.1% for UA. The improvement suggests that the DBN can generate better statistics than the original UBM for standard i-vector space modeling. However, the gain is limited. One potential reason is because the GMM based emotion classifier is not accurate and thus the frame level emotional labels for DBN training are rather noisy. We expect better frame-level predictions can further boost system performance of our proposed DBN-ivector framework.

The following rows in the table show the effect of our two processing strategies to generate reference input for DBN training. In the top-N selection approach, we vary N from $[2, 4, 8]$ out of the 16 mixture components. The best result is achieved when N is 2, i.e., when only the top two Gaussian components are preserved. The WA and UA are improved to 57.2% and 58.6% respectively. When larger N is used, for example, 4 and 8, there is a degradation in both WA and UA, compared to when N is 2. The results are similar to the basic DBN-ivector. When using a smoothing window, we empirically varied K from $[2, 4, 7, 10]$, which means the number of frames used for smoothing is 5, 9, 15 and 21 respectively. We can see that compared to the results of the basic DBN-ivector, smoothing yields some gain when K is not too large. The performance decreases when the window is too big, suggesting that discriminative emotional information may be smoothed out.

Table 2: Accuracy (%)of four emotion categories

Systems		Angry	Happy	Neutral	Sad
DBN-ivector		65.4	50.7	51.6	64.8
DBN-ivector-top2		65.9	51.3	52.6	64.6
DBN-ivector with top K	K=2	65.7	52.5	52.4	62.7
	K=4	66.1	52.1	52.4	62.4
	K=7	65.1	50.3	51.0	66.4
	K=10	63.8	49.4	52.7	65.6

Table 3: Results (%) of decision level combination

System1	System2	WA	UA
DBN-ivector-top2	smoothing K=2	57.6	58.9
	smoothing K=4	58.0	59.3
	smoothing K=7	**58.1**	**59.6**
	smoothing K=10	58.1	59.5

To better understand system performance, the accuracy of each emotion category is shown in Table 2. We can see that when using the top 2 mixtures, performance on Angry and Sad is similar to the basic DBN-ivector, but there is some improvement for Happy and Neutral. When different K is used for smoothing, we observe some different patterns. When K is 2 and 4, both Happy and Neutral have improved performance, but accuracy of Sad is worse. However, when a longer window is used, performance of Sad is significantly improved (it is 66.4% when K is 7). When K increases to 10, there is still some improvement for the Sad category, but the accuracy for Angry and Happy drops. These suggest that we may need different time segments to appropriately represent and model different emotions.

Based on the observation above that systems with top selection and window smoothing for DBN training have gains on different emotion categories, it is worthwhile to treat them as different systems and perform system combination to leverage their strengths. Table 3 shows the decision level combination results using the two approaches. We use the same weight in combination. We combine DBN-ivector-top2 with the smoothing system that uses different window lengths. These results demonstrate that with decision level combination there is a consistent improvement in both WA and UA for all the K values. The best results are obtained when K is equal to 7, 58.1% for WA and 59.6% for UA. Compared to the standard i-vector framework, the improvement is statistically significant (p value < 0.05 with one tailed z-test). This shows that the two systems, with smoothing frames and top N selection strategy for DBN training, complement with each other.

5. Conclusion & Future Work

In this paper, we proposed to combine DBN and i-vector space modeling for acoustic emotion recognition. A trained DBN is used to calculate sufficient statistics for the i-vector framework. Our experimental results show this proposed DBN-ivector method outperforms the standard i-vector framework. We also proposed two post-processing strategies to generate robust labels for DBN-training, and demonstrated that their combination achieves singificantly better performance than the basic DBN-ivector framework. In the future work, we plan to investigate other post-processing approaches to generate more reliable reference labels for DBN training.

6. References

[1] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi *et al.*, "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism," in *Proceedings of INTERSPEECH*, 2013.

[2] F. Ringeval, B. Schuller, M. Valstar, S. Jaiswal, E. Marchi, D. Lalanne, R. Cowie, and M. Pantic, "The AV+EC 2015 multimodal affect recognition challenge: Bridging across audio, video, and physiological data," 2015.

[3] V. Mitra, E. Shriberg, M. McLaren, A. Kathol, C. Richey, D. Vergyri, and M. Graciarena, "The SRI AVEC-2014 evaluation system," in *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*. ACM, 2014, pp. 93–101.

[4] N. Cummins, J. Epps, V. Sethu, and J. Krajewski, "Variability compensation in small data: Oversampled extraction of i-vectors for the classification of depressed speech," in *Proceedings of ICASSP*, 2014, pp. 970–974.

[5] D. Vergyri, B. Knoth, E. Shriberg, V. Mitra, M. McLaren, L. Ferrer, P. Garcia, and C. Marmar, "Speech-based assessment of ptsd in a military population using diverse feature classes," in *Proceedings of INTERSPEECH*, 2015.

[6] V. Mitra, A. Tsiartas, and E. Shriberg, "Noise and reverberation effects on depression detection from speech," in *Proceedings of ICASSP*, 2016, pp. 5795–5799.

[7] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[8] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*, 2014.

[9] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," *Proc. Odyssey-14, Joensuu, Finland*, 2014.

[10] D. Le and E. M. Provost, "Emotion recognition from spontaneous speech using hidden markov models with deep belief networks," in *ASRU, 2013 IEEE Workshop on*, 2013.

[11] R. Xia and Y. Liu, "Using denoising autoencoder for emotion recognition." in *INTERSPEECH*, 2013.

[12] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, 2008.

[13] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[14] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of ICML*, 2010.

[15] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proceedings of ICASSP*, 2013.

[16] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari *et al.*, "Combining modality specific deep neural networks for emotion recognition in video," in *Proceedings of the 15th ACM on International conference on multimodal interaction*. ACM, 2013, pp. 543–550.

[17] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[18] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[19] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.