Interspeech 2018
2-6 September 2018, Hyderabad

# Correlation Networks for Speaker Normalization in Automatic Speech Recognition

*Rini Sharon A, Sandeep Reddy Kothinti, Srinivasan Umesh*

Indian Institute of Technology Madras, India

ee15d210@smail.iitm.ac.in, sandeep.kothinti@gmail.com, umeshs@ee.iitm.ac.in

## Abstract

In this paper, we propose using common representation learning(CRL) for speaker normalization in automatic speech recognition (ASR). Conventional methods like feature space maximum likelihood linear regression (fMLLR) require two pass decode and their performance is often limited by the amount of data during test. While i-vectors do not require two-pass decode, a significant number of input frames are required for estimation. Hence, as an alternative, a regression model employing correlational neural networks (CorrNet) for multi-view CRL is proposed. In this approach, the CorrNet training methodology treats normalized and un-normalized features as two parallel views of the same speech data. Once trained, this network generates frame-wise fMLLR-like features, thus overcoming the limitations of fMLLR/i-vectors. The recognition accuracy using the proposed CorrNet-generated features is comparable with the i-vector model counterparts and significantly better than the un-normalized features like filterbank. With CorrNet-features, we get an absolute improvement in word error rate of 2.5% for TIMIT, 2.69% for WSJ84 and 3.2% for Switchboard-33hour over un-normalized features.

**Index Terms**: Automatic speech recognition, Correlational Neural Networks, fMLLR, Multi-view, Common representation learning, speaker normalization, i-vectors

## 1. Introduction

Recently, there has been an increased interest in investigating speaker normalization techniques to improve the performance of real-time automatic speech recognition (ASR) systems [1–3]. It is widely accepted that speaker normalized features provide gains over un-normalized features [4, 5]. This has set the trend of exploring better methods for extracting speaker normalized features from un-normalized ones. Two very popular methods for speaker normalization are feature space maximum likelihood linear regression (fMLLR) [6, 7] and speaker identity vectors (i-vectors) which carry speaker specific information [8]. However, these conventional methods, suffer from data-insufficiency problems as they require a certain number of frames from a specific speaker for robust estimation. In real-time systems where we encounter short duration utterances from unknown speakers, fMLLR and i-vectors have limitations.

Various feature extraction techniques based on deep neural networks (DNN) are drawing attention lately, of which some methods explicitly perform some form of speaker normalization. Examples of such methods include the pseudo-fMLLR approach [9], which proposes a DNN based feature extractor trained to generate fMLLR-like features from un-normalized filterbank (fbank) features. The pseudo-fMLLR features proved to perform better than handcrafted fbank or MFCC features. Similarly, [10] follows a canonical correlation analysis (CCA) based approach for feature extraction in a multi-view learning framework. Such methods overcome the aforementioned shortcomings of fMLLR/i-vectors.

Given that different kinds of feature representations exist for the same data in different modalities, it would be beneficial to use more than one representation to model our task at hand. This can be achieved by learning common representations from the different feature representations. In CRL approaches, distinct descriptions of data are treated as different parallel views of the same data [11–13]. During training, we may have access to all the views of the data, but while testing certain views may not be available. By learning a common representation between views, we can perform certain tasks such as reconstructing one view from another [14] and improving the performance of a single view system.

Applying the concepts of CRL, correlational neural networks [15] combine two popular techniques, namely, CCA [16] and multi-modal autoencoders (MAE) [17]. CCA is commonly used for learning shared representations of different views of data when they are projected in a highly correlated common space [18–20]. On the other hand, MAE aims to perform self-reconstruction and cross-reconstruction of the parallel views [17]. Hence, combining the complimentary characteristics of the aforementioned approaches, the CorrNets are trained with the dual objective of minimizing reconstruction error and maximizing the correlation between the views in a common projected space. CorrNets have previously been used in the context of image data and text data for the purpose of transfer learning, transliteration and reconstruction of a missing view. One such common multi-view learning application discussed in [15] is to consider the two halves of an image as two views of the same data and reconstruct one view from the other in the case of a missing view.

In this paper, we discuss the first attempt to use CorrNets as a regression based feature extraction module to achieve real-time per-frame speaker normalization for ASR applications. Although our approach follows principles similar to [10], fMLLR normalized features which are speaker independent phoneme representations are used in place of articulatory features. This alleviates the difficulty in obtaining articulatory features for large Databases.

Section 2 in this paper discusses the proposed method for speaker normalization using the CorrNet feature extractor, its architecture and implementation details. Section 3 explains the experiments performed, the data-sets used and the toolkits used for the specific modules. In Section 4, we report the results of all the experiments and discuss the observations that follow. Finally, Section 5 provides a summary of this paper and highlights its contributions.



10.21437/Interspeech.2018-1612

## 2. Proposed Technique

### 2.1. Background

We propose to apply the CRL approach of CorrNets for generating speaker normalized features in order to improve the performance of the speech recognition system for real-time applications. In this framework, for each frame of training data, we assume the availability of two parallel views:

- The un-normalized filterbank or mel frequency cepstral coefficients(MFCC) features as view-1 ($V_1$)
- The normalized fMLLR features as view-2 ($V_2$)

This proposed feature normalization technique deals with the following constraints while testing:

- A single test utterance is available for decoding at each time instant
- Knowledge about the test speaker is unknown

Hence the test utterance has to be treated independently and normalization techniques have to be performed. The CorrNet design is based on the intuition that the normalized and un-normalized features will be correlated in some projected space, as they are just different feature representations of the same data.

### 2.2. Model Architecture and Implementation Details

The speaker normalized features generated by the CorrNet are fed to a DNN Acoustic model for training as shown in Figure 1. The CorrNet feature extractor module is provided with the two input views, $V_1$ as filterbank/MFCC and $V_2$ as fMLLR. In this module, there exists a shared layer resembling a bottleneck layer which is common to both the views. This layer projects the two views into a common space such that the correlation between the views in that space is maximized.

#### 2.2.1. *Objective Function*

The overall optimization loss function for the CorrNet model is as follows:

$$\begin{aligned} Loss = & L_{mse}([none, V_2], V_2^{rec}) \\ & + L_{mse}([V_1, none], V_2^{rec}) \\ & + L_{mse}([V_1, V_2], V_2^{rec}) \\ & - \lambda \times L_{corr}(\mathbb{P}(V_1), \mathbb{P}(V_2)) \end{aligned} \quad (1)$$

where, $L_{ltype}([in1, in2], rec)$ is said to denote the "ltype" loss (mean square loss (MSE) or correlation loss), when the CorrNet is provided with inputs "in1" as view-1 and "in2" as view-2. The MSE is calculated between the original fMLLR feature and the reconstructed fMLLR feature - "rec". $\mathbb{P}(A)$ implies the common layer projection of the input $A$. The scaling factor $\lambda$ is used to adjust the range of correlation loss (corrloss) to match that of reconstruction loss. The negative sign for corrloss implies that we intend to maximize the correlation in the projected space. The corrloss is defined as follows for N input instances:

$$L_{corr}(A, B) = \frac{\sum_{i=1}^{N}(A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_{i=1}^{N}(A_i - \overline{A})^2 \sum_{i=1}^{N}(B_i - \overline{B})^2}} \quad (2)$$

In other words, we state that the CorrNet is trained and optimized to perform the following:

- Reconstruct fMLLR from itself (Self-reconstruction)

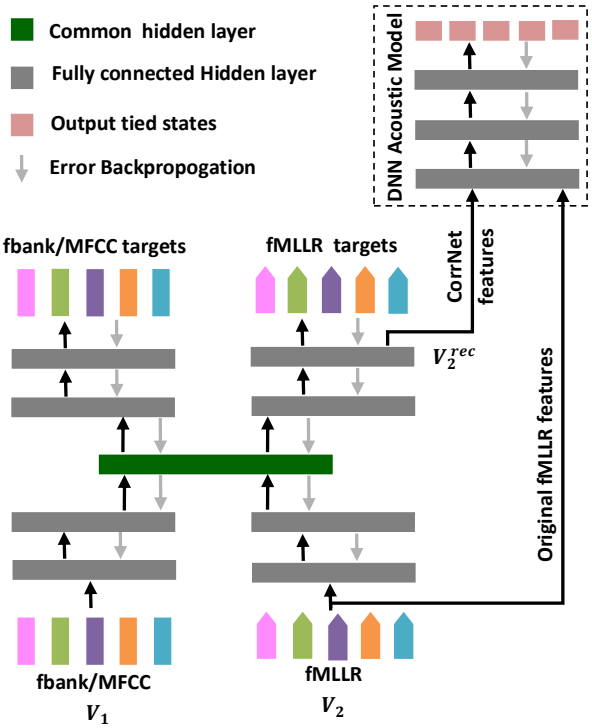


Figure 1: *Proposed Model Architecture*

- Reconstruct fMLLR from fbank/MFCC (Cross-reconstruction)
- Reconstruct fMLLR when both fMLLR and MFCC are provided as inputs (Mixed-reconstruction)
- Maximize the correlation between fMLLR and fbank/MFCC in the common projected space

#### 2.2.2. *Training and Decoding Methodology*

The model should be able to generalize for speakers not seen during training. To achieve this, the validation set is built on unseen speaker identities, in other words, the splitting is done speaker wise and not utterance wise. At test time as depicted in Figure 2 , only un-normalized features are available. Hence these features are passed as inputs to the already trained CorrNet, and their corresponding outputs are taken to be the new normalized features. This process is called projecting the test data through the CorrNet. There are two ways of obtaining the projected representations from the CorrNet,

- Considering the output at the shared common layer as the projected output
- Considering the reconstructed fMLLR output as the projected output

For the task at hand, considering the projection from the reconstructed layer proved to be better. The projected output data is now passed to the DNN acoustic model for decoding.

#### 2.2.3. *Parameters and model-tuning*

There exist two main hyper-parameters that need to be tuned empirically to achieve the best performance using this architecture.

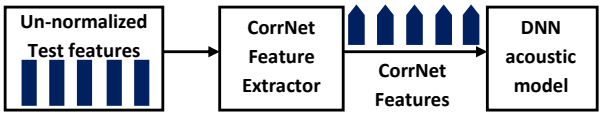


Figure 2: *Decoding Phase*

- Scaling parameter for correlation loss: The range of Correlation loss might be different from the other reconstruction losses. In order to maintain consistency while summing up the losses, a scaling parameter is multiplied to the correlation loss before it is summed up with other losses. This parameter is also tuned to achieve the appropriate scaling for each database.
- Loss weights: Each loss term in the training objective is multiplied by a scalar weight in order to place more emphasis on few of the loss terms as compared to others.

### 2.3. Difference between the proposed method and other existing work

An existing method to generate pseudo-fmllr features [9], proposes that a DNN can learn speaker normalizing transforms like FMLLR, when un-normalized features are fed as inputs and the corresponding speaker-normalized features are provided as targets. The training objective here is only to minimize the mean squared error. In contrast, when we use CorrNets for the same, we provide extra supervision in terms of additional loss terms as formulated in Equation 2. This supervision forces the network to improve its reconstruction abilities while focusing on learning to produce maximally correlated common representations.

CCA based approaches have been previously used for CRL in ASR applications [10]. In that work, the acoustic and articulatory features are taken as two views of the data and the multi-view representation thus learned improves the phonetic recognition. However, articulatory features are tedious to obtain and suffer from data insufficiency issues. Moreover, this approach lacks an explicit reconstruction objective and hence would not be very effective for the purpose of reconstructing a missing view. On the other hand, our approach uses fMLLR as the normalized view and is robust in achieving reconstruction and maximizing correlation.

## 3. Experimental Details

Experiments performed in this section are designed to compare the proposed method with the baselines and other speaker normalization approaches. Input feature extraction was done using the Kaldi toolkit [21] to obtain 13-dimensional MFCC (without delta derivatives), 36-dimensional filterbank and 40-dimensional fMLLR features using a window size of 25ms and a shift of 10ms. Cepstral mean variance normalization (CMVN) was applied to the input features. Two levels of normalization were considered, namely, utterance-norm, where the normalization was done on utterance basis (simulating the real-time constraints) and speaker-norm, where the normalization was done considering per-speaker data.

### 3.1. Details of speech corpus used

TIMIT, WSJ84, and Switchboard-33hr (SWBD-33) subset were used in this paper for testing our proposed model. TIMIT corpus [22] consists of 630 speakers with 10 oral recordings each. 462 speakers' data was taken as the train data, 50 speakers' data was used in development (dev) set and 24 speakers' data was used in test set. A bi-gram language model which was built using the whole train set was used while decoding. In Wall Street Journal speech corpus [23], the training set consists of 7138 sentences and the eval93 subset consists of 213 sentences. Decoding was performed using a tri-gram language model built on the train data. The SWBD-33 considered for this experiment was taken from a 33-hour subset of Switchboard [24]. The evaluation set, also called eval2000 is taken from 2000 HUB5 English evaluation [25] and contains 40 conversations which amount to 2.1 hours of data. The CALLHOME (callhm) [26] set consists of 20 telephone conversations of 30 mins each. The four-gram language model used to decode SWBD-33 was built on the train data.

### 3.2. Details of CorrNet Feature Extractor

Tensorflow toolkit [27] was used to design the CorrNet feature extractor. The input features are spliced to give a 9 frame context as input to the CorrNet. The number of nodes in the shared common layer was taken to be 100 and all other layers in Figure 1 had 512 nodes. Xavier initialization protocol was borrowed from Kaldi to initialize the Tensorflow-CorrNet to maintain a fair comparison of the results. A batch size of 256 with adam optimizer and sigmoid activation function gave the optimal performance in terms of speed and accuracy. The learning rate was set to 0.006 to ensure that there was no variance flooring or Nan values in the output error. The $\lambda$ value was set to 0.5, 2 and 0.1 for TIMIT, WSJ84 and SWBD-33 respectively. The reconstructed output was considered the projected output. All the hyper-parameters were tuned and set to give the best performance intended. Three variations of the CorrNet model were trained.

- CorrNet (Recon fM$\leftarrow$fb) : Here only the reconstruction of fMLLR from fbank/MFCC is kept active during training. This closely resembles the pseudo-fMLLR approach in [9] except for the shared hidden layer which serves as a bottleneck layer.
- CorrNet (All losses) : Here all the losses are active and the CorrNet is trained to minimize the overall objective function.
- CorrNet (Weighted loss) : Here the losses are weighted, so some losses are given more importance than the others.

The combined scoring that was performed on the decode outputs of the three CorrNet models gave the best overall performance.

### 3.3. Details of DNN Acoustic Model

The Kaldi toolkit [21] was used for gaussian mixture modeling (GMM), hidden markov modeling (HMM) and DNN modeling. The alignments generated using the GMM-HMM models trained on fMLLR features were used as the input alignments for the DNN model. The configuration of the DNN acoustic model included a 9-frame input context, 2048 hidden neurons per layer, 3 hidden layers and sigmoid activations. The DNN was pretrained using the layer-wise restricted boltzman machine (RBM) pre-training approach. The cross-entropy training was performed using mini-batch gradient descent with a batch-size of 256. The inputs to the DNN acoustic model during training included the original fMLLR features as well as the

Table 1: *Phone error rate(%) for TIMIT and word error rate (%) for WSJ84 and SWBD-33. Results for CorrNet architectures are reported when filterbank features are provided as inputs.*

| **Input features to DNN Acoustic Model** | **TIMIT** | | | | **SWBD-33** | | | | **WSJ-84** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ***spk-norm*** | | ***utt-norm*** | | ***spk-norm eval2000*** | | ***utt-norm eval2000*** | | ***spk-norm*** | ***utt-norm*** |
| | *test* | *dev* | *test* | *dev* | *swbd* | *callhm* | *swbd* | *callhm* | eval | eval |
| MFCC | 20.3 | 18.9 | 21.4 | 20.0 | 23.7 | 35.5 | 24.5 | 35.7 | 14.26 | 15.86 |
| MFCC + i-vectors | 20.2 | 18.3 | 20.9 | 19.6 | 23.4 | 35.2 | 24.1 | 35.5 | 14.0 | 15.61 |
| Filterbank | 20.0 | 18.4 | 21.4 | 20.7 | 22.8 | 34.6 | 24.3 | 34.6 | 13.74 | 14.96 |
| Filterbank + i-vectors | 19.5 | 17.9 | 21.6 | 19.3 | 22.04 | 33.6 | 23.6 | 34.8 | 13.57 | 14.19 |
| fMLLR | 18.3 | 17.4 | 25.4 | 24.8 | **20.8** | **31.4** | 25.0 | 40.1 | 11.56 | 18.24 |
| fMLLR + i-vectors | **18.1** | **17.1** | 25.1 | 23.8 | 21.02 | 31.4 | 24.7 | 40.1 | **11.36** | 18.04 |
| CorrNet Models | | | | | | | | | | |
| CorrNet (Recon fM←fb) | 19.6 | 18.0 | 19.7 | 18.4 | 21.9 | 33.7 | 21.8 | 35.0 | 12.75 | 13.39 |
| CorrNet (All loss) | 19.4 | 17.9 | 19.0 | 18.3 | 21.53 | 32.5 | 21.9 | 34.5 | 12.64 | 13.29 |
| CorrNet (Weighted loss) | 19.4 | 17.8 | 19.0 | 18.3 | 21.5 | 32.6 | 21.7 | 34.5 | 12.58 | 13.23 |
| Combined Scoring | 18.8 | 17.7 | **18.9** | **18.2** | 21.1 | 32.5 | **21.3** | **34.1** | 12.52 | **13.17** |

CorrNet generated features. For the Baselines, 40-dimensional i-vectors were estimated using a diagonal universal background model GMM (UBM-GMM) trained on input MFCC features. The number of mixture components used for the UBM-GMM were 128 for TIMIT and WSJ84 and 512 for Switchboard 33 hour (SWBD-33) subset.

## 4. Results and Discussions

The results of all the experiments that were performed to validate the proposed model are reported in Table 1. The results are in terms of phone error rate (PER) for TIMIT and word error rate (WER) for WSJ84 and SWBD-33. These error rates are reported for the baselines as well as the CorrNet models discussed in Section 3.2. Both spk-norm and utt-norm results are listed out for the sake of comparison. Utterance wise normalization replicates the real-world scene, where fMLLR, i-vectors and mean-variance normalization are all performed at utterance level. The following observations were made based on the results obtained:

- The performance obtained while using filterbank or MFCC features in isolation to train a DNN classifier is considered as the Baseline. Similarly while performing speaker-wise normalization the performance of fMLLR features or fMLLR + i-vector features are considered as the upper limit performance achievable.
- In the case of MFCC or fbank, adding i-vectors always resulted in superior performance, whereas adding i-vectors to fMLLR did not improve the performance significantly. This may be because the fMLLR features already model the speaker normalizing transforms and may not benefit from the additional speaker information provided by i-vectors.
- Utterance normalization (utt-norm) is what we ideally require for practical applications where we do not get access to adequate data from a specific user. Hence we perform normalization on utterance basis. It is observed across all three databases, that CorrNet models show consistent improvements in the utt-norm scenario by generating frame-wise normalized outputs.
- Speaker-normalization (spk-norm) implies that we have multiple utterances from the same speaker which is used to estimate fMLLR or i-vectors. Even in this case, the CorrNet based model gives performance closest to the best fMLLR model as compared to fbank + i-vector (or) MFCC + i-vector.
- The CorrNet (Recon fM←fb) loss is similar to the pseudo-fMLLR architecture as it contains only one active MSE loss term. However, adding all losses and performing combined scoring further improves the accuracy of the CorrNet based ASR.
- The CorrNet all-loss and weighted-loss models give comparable results across databases. However, while analyzing their decode outputs, we see that they produce different errors. This could justify why combined scoring helps in our case.
- It is observed that performing a combined scoring of the three CorrNet model variants renders further improvements in WER/PER. Combined scoring generates a union of lattices from the input models and then performs a minimum bayes risk decoding on the resulting lattice. The success of score combined can be justified by analyzing the decode outputs of each model. We observe that the decode outputs capture different information when trained with different loss models although the final WER is similar. Therefore, when these models with varying degrees of abstraction are combined, they result in a finer lattice.

## 5. Conclusions

In this paper, we propose a method of speaker normalization for real-time speech recognition applications, where sufficient speaker information is not accessible. A CorrNet feature extraction module is trained to output frame-wise normalized features when provided with un-normalized input features. This proposed multi-view training set up assumes speaker normalized (fMLLR) and un-normalized (MFCC/fbank) features as the two views of speech data available during the training phase. The DNN acoustic models built for TIMIT, WSJ84 and SWBD-33 show that the proposed method of frame-wise speaker normalization give convincing improvements over all DNN baselines including conventional methods like fMLLR and i-vectors for the utterance normalization scenario.

# 6. References


[1] N. M. Joy, S. R. Kothinti, and S. Umesh, "Fmllr speaker normalization with i-vector: In pseudo-fmllr and distillation framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 797–805, 2018.

[2] L. Samarakoon and K. C. Sim, "Learning factorized feature transforms for speaker normalization," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 145–152.

[3] A. Ghoshal, D. Povey, M. Agarwal, P. Akyazi, L. Burget, K. Feng, O. Glembek, N. Goel, M. Karafiát, A. Rastrow *et al.*, "A novel estimation of feature-space mllr for full-covariance models," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4310–4313.

[4] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[5] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7942–7946.

[6] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[7] J. Huang, E. Marcheret, and K. Visweswariah, "Rapid feature space speaker adaptation for multi-stream hmm-based audio-visual speech recognition," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 338–341.

[8] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*, 2013, pp. 55–59.

[9] N. M. Joy, M. K. Baskar, S. Umesh, and B. Abraham, "Dnns for unsupervised extraction of pseudo fmllr features without explicit adaptation data." in *INTERSPEECH*, 2016, pp. 3479–3483.

[10] R. Arora and K. Livescu, "Multi-view cca-based acoustic features for phonetic recognition across speakers and domains," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7135–7139.

[11] K. M. Hermann and P. Blunsom, "Multilingual distributed representations without word alignment," *arXiv preprint arXiv:1312.6173*, 2013.

[12] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning: objectives and optimization," *arXiv preprint arXiv:1602.01024*, 2016.

[13] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *International Conference on Machine Learning*, 2013, pp. 1247–1255.

[14] G. Bhatt, P. Jha, and B. Raman, "Common representation learning using step-based correlation multi-modal cnn," *arXiv preprint arXiv:1711.00003*, 2017.

[15] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran, "Correlational neural networks," *Neural computation*, vol. 28, no. 2, pp. 257–285, 2016.

[16] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[18] J. Benesty and I. Cohen, *Canonical Correlation Analysis in Speech Enhancement*. Springer, 2018.

[19] O. Yair and R. Talmon, "Multimodal metric learning with local cca," in *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. IEEE, 2016, pp. 1–5.

[20] H. Sagha, J. Deng, M. Gavryukova, J. Han, and B. Schuller, "Cross lingual speech emotion recognition using canonical correlation analysis on principal component subspace," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5800–5804.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[22] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[23] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[24] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 517–520.

[25] L. D. Consortium, "2000 hub5 english evaluation speech ldc2002s09," *Web Download*, 2002.

[26] A. Canavan, D. Graff, and G. Zipperlen, "Callhome american english speech," *Linguistic Data Consortium*, 1997.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.