NOVEL NEURAL NETWORK BASED FUSION FOR MULTISTREAM ASR

Sri Harish Mallidi[1], *Hynek Hermansky*[1,2]

[1]Center for Language and Speech Processing &
[2]Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, U.S.A

ABSTRACT

Robustness of automatic speech recognition (ASR) to acoustic mismatches can be improved by multistream framework. Frequently used approach to combine decisions from individual streams involve training large number of neural networks, one for each possible stream combination. In this work, we propose to simplify the fusion by replacing the large number of fusion networks with a single fusion network. During training of the proposed fusion network, features from a stream are randomly dropped out. At test time, corrupted streams are identified and dropped out to improve robustness. Using the proposed approach, we were able to achieve significant reduction in number of parameters, while remaining in less than 2.5 % relative degradation of conventional fusion technique. Furthermore, proposed fusion network is also applied in a multistream ASR system to improve noise robustness of Aurora4 speech recognition task. Noticeable improvements were observed over baseline systems (relative improvement of 9.2 % in microphone mismatch and 3.2 % in additive noise conditions).

Index Terms— multistream ASR, performance monitoring, stream fusion, deep neural networks

1. INTRODUCTION

Performance of automatic speech recognition (ASR) technology improved significantly with emergence of deep neural network (DNN) models. Despite huge improvements, the technology is sensitive to acoustic mismatch between train and test data sets. Robustness to acoustic mismatches can be improved using multistream automatic speech recogntion (ASR) framework [1, 2].

Multistream framework involves constructing multiple parallel information processing streams, where each stream is attending to different part of the signal space, and adaptive fusion of decisions from the streams [3]. The fundamental motivation behind multistream recognition is, noise or environmental distortion effects only few parts of the signal space (e.g. frequency bands or spectro-temporal freqency bands). The portions which are less corrupted can be identified and decisions from these streams can be emphasized.

This work was supported in parts by the National Science Foundation via award number IIA-0530118, .Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013, by Google via Google faculty award to Hynek Hermansky. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Google, NSF, IARPA, DoD/ARL, or the U.S. Government.

In order to build a multistream system one needs to deal with the following issues [3]: **Formation of streams:** streams which are conditionally independent as much as possible, while retaining some cues for recognition of message. **Identification of streams:** accurate identification of streams that are less corrupted for a given test signal. **Fusion of streams:** finding combination of streams which gives the best result.

Many previous implementations of multistream system used streams which are localized to few frequency bands [4, 5, 6, 7]. More general ways forming streams have also been studied. These include streams covering different portions of temporal modulations [8, 9], spectro-temporal (2-D) modulations [10, 11, 12], etc. Several techniques have been proposed to identify robust streams [6, 13, 14]. These are primarily based on output of neural network classifiers. The next stage of the processing involves fusion of decisions from the streams. Hermansky et. al. [2] investigated various fusion strategies and found out that neural network based fusion is most effective. Application of neural network fusion in multistream is done by formation all non-empty combinations of features/decision from streams, and training a separate neural network for each combination. This results in large number of fusion networks. For example, [2, 15] used a 7 sub-band system, which requires $2^7 - 1 = 127$ fusion networks. The large number of neural networks can increase the complexity of the system, making training and testing time consuming. It can also deter practical applicability of multistream system.

In this paper, we attempt to reduce the complexity of the fusion stage. Multiple neural networks used in fusion stage are replaced by a single neural network. Input to proposed fusion network is formed by concatenation of features from the streams. During test, features from the corrupted streams needs to be discarded. This can achieved by dropping out (i.e. multiplyling with zero) decisions/features from the corrupted streams. A test vector having portion of its input dimensions as zeros can break down the fusion network. We hypothesize that, the breaking down can be avoided if: during training, the network sees partially zeroed out input vectors. We show that this can be achieved by training fusion network with randomly switching-off features from a stream.

The rest of the paper is organized as follows. In section 2, multistream ASR architecture used in previous works is described. Section 3 describes the proposed fusion network. Section 4 presents the comparison study of proposed fusion network. Noise robust experiments are presented in section 5. In section 6, we conclude with a brief discussion of the proposed method.

2. MULTISTREAM ASR

Figure 1 depicts the block diagram of a multistream system, similar to the one used in [15]. For simplicity, we illustrated a system with

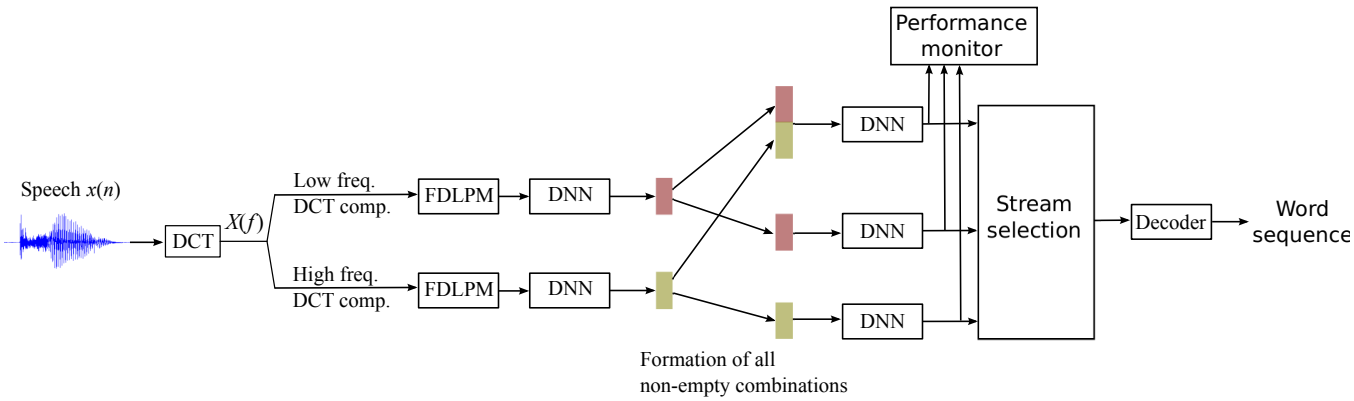

Fig. 1. Block diagram of past multistream system, based on 2 sub-bands. The architecture is similar to 7 sub-band architecture used in [2, 15]

2 streams. The first stage involves forming band-limited streams. Speech signal is divided into two sub-bands, covering low-frequency and high-frequency portion of the signal. Acoustic features corresponding to the streams are extracted and a DNN is trained in each stream.

The second stage involves fusion of decisions from streams of the first stage. For a test signal, the fusion module should evaluate reliability of all streams, and perform combination such that information from reliable streams are amplified. Linear fusion studied in [2, 16, 13, 17] involves computing a weighted average of phoneme (or log-phoneme) posterior probabilities, where the weights are proportional to reliability of streams. Linear fusion is a simple strategy, but it does not account for complementary information present in streams. In order to achieve this, [2] proposed to use neural networks for non-linear fusion of streams. Neural network fusion involves constructing all possible nonempty combinations of decisions from first stage streams, and training a DNN on each combination. This results in a significant increase in complexity. The massive number of fusion networks can deter the application of multistream system to large scale scenarios and practical scenarios. In this work, we propose a new fusion approach which involves training a single fusion network. From here on we refer to fusion using multiple neural networks as $Multi\_Nnet\_fusion$.

3. PROPOSED FUSION NETWORK

We propose to replace multiple fusion networks present in full combination multistream system with at single neural network. Input feature vector to the proposed fusion network is formed by stacking features from first stage processing. During test, when some of the streams are corrupted, robustness can still be retained by dropping out (i.e. multiplying with zero) feature vectors from corrupted streams. The fusion neural network can break down if it sees a test vector with some of its elements as zeros. We avoid this by randomly dropping out features of a stream during training. This forces the network to learn to classify even when part of test vector is zero. The proposed training procedure is illustrated in figure 2. Each stream is dropped-out with probability equal to 0.5. Proposed fusion technique is referred to as *Single_Nnet_fusion*.

The proposed approach is motivated from dropout learning used to regularize DNNs. The differences between dropout learning [19] and proposed training technique are:

- Dropout is applied to all neurons units in all layers (input and hidden). In the proposed technique dropout is applied only to input layer.
- Input neurons belonging to a stream are dropped out together. In dropout each neuron is independently dropped out.

The resulting fusion network can be seen as training a collection of $2^N - 1$ neural networks with weight sharing. During test, we employ a performance monitoring technique to identify combination of streams which gives lowest error rate.

4. COMPARISON EXPERIMENTS

In this section, we compare *Single_Nnet_fusion* with *Multi_Nnet_fusion*. Aim of the experiments to study whether we can achieve word-error-rates (WERs) close to *Multi_Nnet_fusion*, using *Single_Nnet_fusion*.

4.1. Experimental setup

We used Aurora4 [21] database to compare *Multi_Nnet_fusion* and proposed *Single_Nnet_fusion*.

Aurora4 task is a small scale (14 hour), medium vocabulary speech recognition task, aimed at improving noise and channel robustness. The database is based on the DARPA Wall Street Journal (WSJ0) corpus which consist of clean recordings of read speech, with 5000 word vocabulary size. The training set consists of 14 hours of clean speech from 83 speakers, sampled at 16 kHz. The test set consist of 330 recordings from 14 conditions. Each condition include clean testing with same microphone, clean testing with different microphone, 6 additive noise conditions which include airport, babble, car, restaurant, street and train noise at 5-15 dB signaltonoise ratio (SNR) and 6 conditions with the combination of additive and channel noise.

Architecutre of DNN models used in the present work consist of 4 hidden layers. Each hidden layer consist of 1024 Simoidal neurons. Context dependent tri-phone targets are used to as targets while training DNN models. The targets are generated using a HMM-GMM system trained on MFCC features. HMM-GMM system is implemented using Kaldi speech recognition toolkit [22], and DNN models are trained using Theano toolkit [23, 24].

The acoustic feature used to train DNNs at the first stage of multistream system are based on Frequency-domain linear prediction (FDLP) processing of speech. The time-domain signal is transformed into frequency domain by using a discrete cosine transform

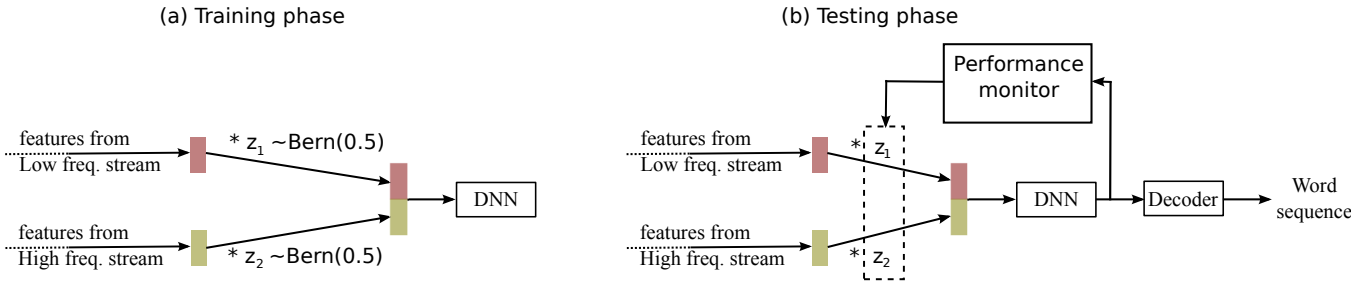

Fig. 2. Illustration of multistream system with proposed fusion network. (a) During training, features from a stream are randomly dropped-out with probability 0.5. (b) During testing, streams are dropped-out deterministically by performance monitor techniques.

Table 1. Comparison of proposed fusion and past fusion approach, for 2-stream system.

	LF+HF	LF	HF
Multi_Nnet_fusion	4.95	8.39	12.07
Single_Nnet_fusion	5.06	8.56	13.19

Table 2. Relative difference between WERs of multistream systems implemented using *Multi_Nnet_fusion* and *Single_Nnet_fusion*.

2-streams	5-streams	7-streams
4.67	11.34	13.92

(DCT) transform. Sub-band DCT coefficients are computed by windowing the full-band DCT signal. Hilbert envelopes of sub-band signals are estimated using linear prediction of DCT signal, resulting in FDLP envelopes. In each sub-band, the estimated envelopes are log-compressed and temporal modulation coefficients corresponding to 0-35 Hz modulation frequencies are computed. These are used to train DNN models used in the present work.

4.2. 2-streams:

A 2-stream system is constructed by using sub-bands covering first three critical bands as low-frequency stream and next two critical bands as high-frequency stream. FDLPM features from the streams are used to multistream system. We first show the comparison results on clean test set of Aurora4 database. Table 1 show the comparison between WER values of multistream systems implemented using *Multi_Nnet_fusion* and *Single_Nnet_fusion*. The second column (under LF+HF) of table 1 represent WER of clean test set obtained when both the streams are used. That is: in the case of *Multi_Nnet_fusion*, WER value obtained from fusion network trained on feature vectors obtained by stacking features from both low and frequency streams. In the case of *Single_Nnet_fusion*, when both the streams are switched-on. The third column represent WER values obtained when only low-frequency stream is used. That is, in the case of *Multi_Nnet_fusion*, WER value obtained from fusion network which is trained only on low-frequency stream. In the case of *Single_Nnet_fusion*, WER value obtained by switching-on low-frequency stream and switching-off high-frequency stream. Similarly, fourth column represent WER value obtained by using only high-frequency stream. From table 1, it is evident that WER values from *Single_Nnet_fusion* are close to *Multi_Nnet_fusion* architecture. We reduced number of parameters by a factor of 3 with an average absolute degradation of 0.46 % and average relative degradation of 4.67 % in performance.

4.3. More streams:

Performance of *Single_Nnet_fusion* in 5-stream and 7-stream cases is analyzed in this section. Table 2 show the degration observed by employing *Single_Nnet_fusion* instead of *Multi_Nnet_fusion* in 2, 5, and 7 sub-band systems. It is evident from the table that, as we increase the number of streams, perforamance of *Single_Nnet_fusion* is decreasing. We hypothesize the reason for this degradation is relatively fewer number of examples of a particular of input vector type is seen during training. In the case of 2-stream system, the fusion network sees one-third of times full feature vector, one-third of times feature vector with high frequency stream dropped out, and the other one-third of times low frequency stream dropped. Whereas in case of 5-stream and 7-stream systems, each input type is seen only 1/31 times and 1/127 times, resepectively. This issue can be circumvented by presenting the training data multiple number of times, resulting in more examples of a input type. Relative degradation with number of iterations through training data per epoch is shown in figure 3. It is evident from the figure that going through training data multiple number of times per epoch increases performance of *Single_Nnet_fusion*, supporting our hypothesis. With 2-3 iterations per epoch, we were able to achieve performance close to 5 % of *Multi_Nnet_fusion*, with a significant decrease in the number of parameters, (1/31 in 5-stream case and 1/127 in case of 7-stream case). In the case of 2-streams, the performance after 5 iterations might be due to overfitting of neural network on the training data. This shows that the choice of number iterations depends on number of streams, as well as on the amount of training data.

5. NOISE ROBUST EXPERIMENTS

In this section, we compare noise robustness of multistream system with $Single\_Nnet\_fusion$ with various other baseline features. Similar to previous section, we use Aurora4 clean training set to train the models. The test set consist of original Aurora4 test set (clean and noise) and an artificially corrupted test test. The artificial test set consist of two band-pass filtered exhibition hall noises from NOISEX database. They are designed to corrupt streams covering the low-frequency portion of the signal, i.e. streams 1 and 2. These are added at -20 dB signal-to-noise ratio to clean test set. This guarantees at least few of the streams remain uncorrupted by noise, satisfying the basic premise of multistream system.

We used a HMM-DNN system trained on Mel filter bank energies (MFBE) as one of the baseline systems. The MFBE system is trained on 40 dimensional filter bank energies, with a 21 frame

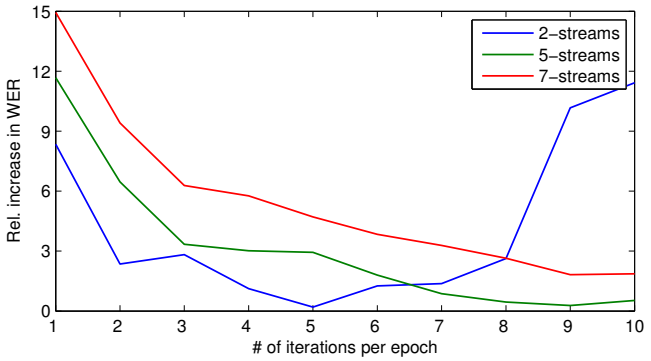

Fig. 3. *Single_Nnet_fusion* can be further improved by presenting training data multiple number of times per epoch. Relative difference between *Single_Nnet_fusion* and *Multi_Nnet_fusion* with number of iterations per epoch.

context. Speaker based mean normalization is applied to the input features. In order to analyze the effectiveness of multistream system, we also compare with singlestream system based on FDLPM features (SS_FDLPM).

For the multistream system, we train a 5-stream system similar to the one described in previous section. The 5-streams consist of sub-bands covering 15 critical bands (each stream covers 3 critical bands). FDLPM features extracted from each sub-band and are used to train a DNN. TANDEM features are extracted for each stream from the first stage DNNs. The TANDEM features are concatenated to form feature vector for the second stage DNNs. The singlestream system (SS_FDLPM) shown in table 3 models the concatenated in the standard way.

We use the technique proposed in Sec. 3 to train a multistream system. Input representation for the multistream system is concatenated feature vector from the first stage TANDEM features. During test time, we use M-delta measure [25, 26] to estimate quality of posteriors of each stream combination.

M-delta performance monitor: Main idea behind the measure is as follows: posterior vectors belonging to the same class should have smaller divergence than the divergence between posterior vectors belonging to different classes. M-delta measure is defined as $\mathcal{M}delta = \mathcal{M}^{\mathrm{ac}} - \mathcal{M}^{\mathrm{wc}}$, where $\mathcal{M}^{\mathrm{ac}}$ and $\mathcal{M}^{\mathrm{wc}}$ represent the accumulated KL-divergence computed from a data pair from the same class and that from a data pair from different classes, respectively. $\mathcal{M}^{\mathrm{ac}}$ and $\mathcal{M}^{\mathrm{wc}}$ are estimated from $\mathcal{M}$ measure [20]. For a test utterance, an M-delta measure is computed for each stream combination. Posteriors from stream combination having highest M-delta measure are selected, converted into pseudo-log-likelihoods and given as input to recognizer.

It is evident from table 3 that multistream system performs better than MFBE system in all the conditions. In the synthetic noisy conditions, synth1-20dB and synth2-20dB, performance is significantly better than single stream systems. This shows that, when the noises satisfy the assumption of multistream system, we can achieve significant robustness. It performs better than SS_FDLPM in most of the conditions, except Babble and Restaurent. This might be due to wide-band nature of these noises, where all the streams get corrupted and none of the combination matches with training statistics. We also observe noticeable improvements in microphone mismatch conditions. The improvement is significant compared to MFBE. This might be due to robustness of FDLPM features. The improvement over SS_FDLPM system in microphone mismatch conditions could be due accurate selection of good combination by performance monitoring.

Table 3. Word error rate (%) in synthetic band-limited noises and Aurora-4 test sets, using various features. The systems are trained using clean training test.

	MFBE	SS_FDLPM	Multistream with proposed fusion
Synthetic noise corrupting only low-frequencies			
synth1-20dB	78.68	60.99	**9.43**
synth2-20dB	80.78	20.51	**14.96**
Aurora4 noises			
Clean Same Mic	**3.75**	4.89	4.41
Clean Diff Mic	17.35	14.53	**12.98**
Additive Noise Same Mic			
Airport	39.60	34.17	**31.29**
Babble	44.09	**39.08**	39.19
Car	20.19	15.47	**13.24**
Restaurant	46.24	**36.11**	36.61
Street	54.79	45.82	**44.95**
Train	51.52	45.10	**43.45**
Avgerage	42.73	35.96	**34.79**
Additive Noise Diff. Mic			
Airport	52.03	45.90	**40.16**
Babble	54.14	51.75	**46.39**
Car	32.54	28.10	**22.29**
Restaurant	57.65	48.42	**46.10**
Street	61.95	54.96	**51.99**
Train	59.01	55.54	**51.43**
Average	52.88	47.45	**43.06**

6. CONCLUSIONS

In this paper, we simplified neural network fusion in multistream speech recognition framework. The proposed fusion network is trained by randomly dropping out features from streams. This forces the network to learn to classify even when some of the streams are dropped out. During test, the choice of streams to drop is determined by performance monitor technique. The proposed technique significantly reduces the number of parameters used conventionally, by replacing multiple neural networks in fusion stage with a single neural network. This resulted in parameter reduction by factors of 3, 31 and 127 in 2-stream, 5-stream and 7-stream cases, respectively. The proposed techinque is with in 3 % relative degradation in 2-stream and 5-stream cases and 7.5 % relative degradation in 7-stream case, compared conventional fusion which involves multiple neural networks. We have also shown that the degradation can be further reduced, by presenting the triaining data multiple number of times per epoch.

We implemented a 5 sub-band multistream system, with proposed fusion network in noise robust ASR task. For frequency localized noises, which satisfy multistream assumption, significant improvements were observed. For wide-band noises, we did not observed any improvements over single stream FDLP features. This is expected since, any combination of input streams does not result in matching with training statistics of output posterior vectors. Robustness in these noisy conditions can be improved by using more general streams which are localized in spectro-temporal space. The proposed fusion method can be applied to these streams as well.

7. REFERENCES

[1] H. Bourlard, S. Dupont, H. Hermansky, and N. Morgan, "Towards subband-based speech recognition," in *Proc. EUSIPCO*, 1996, pp. 1579-1582.

[2] H. Hermansky, S. Tibrewala, and M. Pavel, "Towards ASR on partially corrupted speech," in *Proc. Int. Conf. Spoken Lang. Process.*, 1996, pp. 462-465.

[3] H. Hermansky, "Multistream recognition of speech: Dealing with unknown unknowns," Proc. *IEEE*, vol. 101, no. 5, pp. 1076-1088, May 2013.

[4] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and re-combination of partial frequency bands," in *Proc. Int. Conf. Spoken Lang. Process.*, 1996, pp. 426-429.

[5] S. Tibrewala and H. Hermansky, "Multi-stream approach in acoustic modeling," in *Proc. DARPA Large Vocabulary Cont. Speech Recognit. Hub 5 Workshop*, 1997, pp. 1255-1258.

[6] S. Tibrewala and H. Hermansky, "Sub-band based recognition of noisy speech," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 1997, pp. 1255-1258.

[7] A. Morris, A. Hagen, H. Glotin, and H. Bourlard, "Multi-stream adaptive evidence combination for noise robust ASR," *Speech Commun.* vol. 34, no. 1-2, pp. 25-40, 2001

[8] H. Hermansky and S. Sharma, "TRAPS, classifiers of temporal patterns," in *Proc. Int. Conf. Spoken Lang. Process.*, Sydney, Australia, 1998, Paper 615

[9] H. Hermansky, and P. Fousek, "Multi-resolution rasta filtering for TANDEM-based ASR." in *Proc. Interspeech* 2005.

[10] B. T. Meyer and B. Kollmeier, "Optimization and evaluation of Gabor feature sets for ASR," in Proc. *Interspeech*, 2009, pp. 906-909.

[11] M. Kleinschmidt and D. Gelbart, "Improving word accuracy with Gabor feature extraction," in Proc. *Int. Conf. Spoken Lang. Process.*, 2002, pp. 25-28.

[12] N. Mesgarani, S. Thomas, and H. Hermansky, "Adaptive stream fusion in multistream recognition of speech," in *Proc. Interspeech*, 2011, pp. 2329-2332.

[13] S. Okawa, E. Bocchieri, and A. Potamianos. "Multi-band speech recognition in noisy environments." Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 2. IEEE, 1998.

[14] H. Misra, H. Bourlard, and V. Tyagi. "New entropy based multi-stream combination." In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). IEEE International Conference on, vol.1, pp. I-193. IEEE, 2004.

[15] E. Variani and H. Hermansky, "Estimating classifier performance in unknown noise," in Proc. Interspeech, 2012, Paper 584.

[16] J. Kittler and M. Hatef, "On combining classifiers," IEEE Trans. on Pattern Analysis and Machine Intelligence, 1998, vol. 20, pp. 226-239,

[17] S. H Mallidi, et. al. "Autoencoder Based Multi-Stream Combination for Noise Robust Speech Recognition", in Proc. Interspeech, 2015.

[18] S. Tibrewala, and H. Hermansky. "Sub-band based recognition of noisy speech." Acoustics, Speech, and Signal Processing, IEEE International Conference on. Vol. 2. IEEE Computer Society, 1997.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting," JMLR, 2014.

[20] Hermansky, Hynek, Ehsan Variani, and Vijayaditya Peddinti. "Mean temporal distance: Predicting ASR error from temporal properties of speech signal".Acoustics, Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[21] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR Evaluation," Technical Report, 2002.

[22] D. Povey, A. et. al., "The Kaldi speech recognition toolkit," in Proc. IEEE ASRU, December 2011.

[23] F. Bastien, et. al. "Theano: new features and speed improvements," NIPS 2012 deep learning workshop.

[24] J. Bergstra, et. al. "Theano: A CPU and GPU Math Expression Compiler," Proc. Python for Scientific Computing Conference (SciPy) 2010. June 30 - July 3, Austin, TX

[25] H Hermansky et. al. "Towards machines that know when they do not know: Summary of work done at 2014 Frederick Jelinek Memorial Workshop," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015

[26] Sri Harish Mallidi, T. Ogawa, and H. Hermansky "Uncertainty estimation of DNN classifiers", in Proc. IEEE ASRU, December 2015.