Interspeech 2018
2-6 September 2018, Hyderabad

Stream Attention for Distributed Multi-Microphone Speech Recognition

Xiaofei Wang[1,2], *Ruizhi Li*[1], *Hynek Hermansky*[1]

[1]Center for Language and Speech Processing, Johns Hopkins University
[2]Institute of Acoustics, Chinese Academy of Sciences

xwang176, rli33, hynek@jhu.edu

Abstract

Exploiting multiple microphones has been a widely-used strategy for robust automatic speech recognition (ASR). Particularly, in a general hands-free scenario, acquisition of speech usually happens using a set of distributed microphones or arrays simultaneously. Each microphone or array (defined as a stream) carries a different quality of information. The technique of stream fusion is beneficial to provide the best distant recognition performance against the effects of potential disturbances such as noise, reverberation, as well as the speaker movement.

In this work, we propose a stream attention framework to improve the far-field ASR performance in the distributed multi-microphone configuration. Frame-level attention vectors have been derived by predicting the ASR performance of the acoustic modeling of individual streams using the posterior probabilities from the classifier. They are used to characterize the amount of useful information each stream contributes, for the purpose of an efficient and better-performing decoding scheme. In this paper, we investigate the ASR performance measures using our proposed stream attention system on real recorded datasets, Mixer-6 and DIRHA-WSJ. The experimental results show that the proposed framework yields substantial improvements in word error rate (WER) compared to conventional strategies.

Index Terms: Distributed multi-microphone ASR, stream attention, performance monitor, Posterior probability distribution.

1. Introduction

Hands-free far-field speech recognition in real environments has received a great deal of interest in the speech recognition community. Making the recognizer robust to noise and reverberation has been a great challenge. In far-field ASR scenarios, it is feasible to use many parallel recognition streams. Recognition of speech from multiple acoustic streams obtained from microphones distributed in space is a situation that needs to be solved. Depending on the room situation and microphone status, some streams (microphones closer to the speaker, less noise and reverberation, more matched with the training data) may deliver better recognition results than the others. In such a situation, automatically selecting the best microphone for ASR, and further achieving a potential better ASR performance through combining the microphones is desirable. Conventional solutions such as selecting the acoustic stream with the highest energy are vulnerable to strong noises [1].

There are several ways to enhance the ASR performance utilizing the multi-microphone configuration. One possible strategy is to align the time delay between the microphones and use spatial information to carry out beamforming at the signal level [2][3]. However, in the distributed setup, time delays are difficult to estimate [4]. As a front-end processing module, the objective functions used in beamforming are also not optimal for ASR [5]. Another way of approaching this problem is to find the highest likelihood combination of best paths through multiple recognition lattices, formed from all individual streams [6][7]. This requires carrying out full searches in each microphone stream, which is typically done over the whole length of each utterance. The difficulty with this approach is the computing complexity of the multiple decoding operations.

Most ASR systems require feature vectors, which represent information about underlying speech sound at regular time intervals. Such feature vectors can be derived from posterior probabilities of the sounds, estimated by deep neural network (DNN) classifiers. DNN posteriors are able to tolerate the misalignment between the classifier inputs and corresponding labels [8]. We propose to construct at every time instant the best feature vector from a combination of the most reliable sound posteriors from different available streams, which is defined as a stream attention scheme. Hence, in our setup, only one decoding operation for ASR is needed, which is computationally more beneficial than multiple decoding operations. Specifically, an attention scheme can be achieved by generating an attention vector for multiple inputs [9][10], where the attention vector plays the important role of addressing the crucial parts of the inputs. Given the feature vectors (DNN posteriors), the key problem of stream attention is to find an appropriate measure of the goodness for the feature vectors in the individual streams. This goodness measure could then be used in deriving a proper attention vector for the construction of the best feature vector [11][12][13][14][15][16].

Inspired by these mechanisms, we propose a stream attention framework to improve the distributed multi-microphone ASR performance. The unsupervised ASR performance monitor (PM) is used to build the relationship between the goodness of DNN posterior vectors and the ASR performances to calculate the discriminative attention vectors. Consequently, for each time instant, we construct the signal acquisition with the best fusion of the most reliable microphones or arrays using the attention vector, which is applied to the DNN posteriors derived from all the streams. Further, we extend the auto-encoder based PM [16] by including the temporal information which is a distinctive property of speech. Based on detailed ASR experiments using the real datasets, we achieved a robust performance in several representative well-designed scenarios.

The remainder of this paper is organized as follows: Section 2 describes the proposed stream attention framework of the distributed multi-microphone system. In section 3, different ASR performance measures are compared via ASR experiments using real recordings. Section 4 concludes the paper.

The work is supported by National Natural Science Foundation of China (No.61601453) and a Google faculty award to Hynek Hermansky.

10.21437/Interspeech.2018-1037

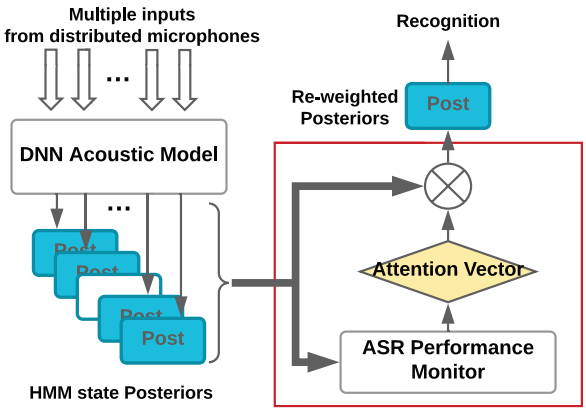

Figure 1: *Stream attention framework using the posterior probabilities from DNN classifier in the distributed multi-microphone setup. The inputs can be signals from individual microphones or the synthesized signals from beamformings.*

2. Proposed Framework

In this section, we describe the stream attention framework applied to the posterior probabilities over Hidden Markov Model (HMM) states to force the recognizer to automatically focus on the more reliable microphones. The diagram in Fig.1 demonstrates the attention scheme and attention vector estimation (red rectangular in Fig.1) using multiple posteriors - each obtained from the corresponding Softmax output of a typical DNN-HMM classifier.

2.1. Formulation of the Stream Attention Scheme

As suggested by Fig.1, let $\mathbf{P}_t = [P_t^1, P_t^2, ..., P_t^M]^T$ denote the posterior probability sequences of HMM states O at time t, where ^{T} is the transpose operation and $P_t^i = p(O|\mathbf{X}_t^i), i = 1, ..., M$ is the ith posterior probability sequence given the feature sequence $\mathbf{X}_t^i$ extracted from the signal of microphone i. M is the total stream number, which is equal to the number of microphones (or arrays). Specifically, $\mathbf{X}_t^i = [X_{t-\tau}^i, ..., X_t^i, ..., X_{t+\tau}^i]^T$ is context based, including $2\tau + 1$ adjacent frames centered at time t.

Assuming that we have the stream attention vector $\mathbf{w}_t = [w_t^1, w_t^2, ..., w_t^M]^T$, which is an M-element vector with summation equal to 1 at time t, we are able to achieve the re-weighted posterior probability sequence $\hat{P}_t$ as follows,

$$\hat{P}_t = \mathbf{w}_t \mathbf{P}_t \tag{1}$$

After the re-weighted combination, $\hat{P}_t$ is used for decoding.

2.2. ASR Performance Monitor informed Attention Vector

For each time instant, the attention vector is estimated by evaluating the relative ASR performance between the microphone streams in an unsupervised way.

2.2.1. Entropy of the phoneme posterior distribution

Researchers proposed to distinguish the ASR performance in each stream by observing the relationship between recognition accuracy and the phoneme posterior distribution. The posterior distribution at a particular time point would converge to non-informative, as the signals were increasingly corrupted by noise or reverberation. Therefore, inverse entropy $1/H_i$ of $\overline{P}_t^i$ is a measure to determine the performance of microphone stream i [11][12], so that the attention vector of each frame is given by

$$w_t^i = \frac{1/H_i}{\sum_{i=1}^{M} 1/H_i} \tag{2}$$

2.2.2. Long-term M-measure using the posteriorgram

By considering the temporal properties of phoneme posterior probability, a mean time distance (M-measure) accumulates how similar or different every two probability vectors $P_{t-\Delta t}^i$ and P_t^i are, by calculating their symmetric Kullback-Leibler divergence $D(P_{t-\Delta t}^i, P_t^i)$ spaced over several time-spans [13][15]. If the speech were corrupted by stationary or slowly varying distortions, these distortions start dominating the signal and the phoneme posteriors become more similar, resulting in a lower average value of M-measure. M-measure relies on long-term windows over hundreds of milliseconds. Stream with better ASR performance would have a larger value than the other streams in this long-term window. Thus, a time-invariant attention vector having binary elements across the window is derived, which is given by $w_t^i == 1$, if $M^i(\Delta t) > M^j(\Delta t)$, where $i \neq j$, t belongs to all the frame times in the window.

2.2.3. Extended auto-encoder with temporal context training

The multi-layer neural network is good at modeling the complex data distributions. An auto-encoder can be used as an ASR PM to model the output activations of DNN acoustic model [16].

Inspired by M-measure using the temporal dynamics to predict the ASR performance, in this study, we extend the auto-encoder PM [16] through training with the context-based posterior features centered by the current frame as the input, and current frame at time t as the training target. To further relax the strict alignment of input features and corresponding targets and significantly reducing the input size, we exploit the time-delayed neural network (TDNN) structure with splices in the hidden layers to train the auto-encoder [17].

Specifically, in the training phase, the auto-encoder is trained on the HMM state posterior sequences with Logit (to make the features more Gaussian) and a principal component analysis (PCA) transformation (transformation basis of PCA is evaluated from the training data). The data for training the auto-encoder is the same as that for training the DNN classifier. Mean square error (MSE) criterion is used as the cost function for auto-encoder training. In the test phase, the reconstruction error of test data is used as a measure of stream confidence, which means that a vector similar to the distribution of training data will yield a low reconstruction error compared to vectors drawn from a different distribution. The lower the reconstruction error is, the better test and training data are matched, resulting in a better recognition accuracy. Therefore, an auto-encoder based frame-wise attention vector element w_t^i can be derived as follows

$$w_t^i = \frac{1/||e_i||^2}{\sum_{i=1}^{M} 1/||e_i||^2} \tag{3}$$

where $||e_i||$ is the l_2 norm of reconstruction error vectors.

3. Experiments and Results

3.1. Dataset and Baseline

The framework coupling with PMs was evaluated on two recorded datasets, which are a subset of Mixer-6 dataset [18] and DIRHA-WSJ dataset [19].

3.1.1. Mixer-6 dataset (without speaker movement)

This dataset consists of a set of US English speakers reading a list of sentences. The recordings were conducted on-site by Linguistic Data Consortium in two distinct office rooms (denoted by "LDC" and "HRM" room) equipped with multi-channel recording platforms. Each room was set up with a matching set of 13 distinct microphones, placed at equivalent locations relative to the speaker. However, the speaker did not move during recording. The transcribed dataset was separated into training part and testing part for ASR experiments. For each utterance, we had thirteen synchronous (but not time-aligned due to the delay in propagation of the sound wave) recordings simultaneously. We used the recordings from microphone 2 (head-mounted microphone, best acoustic channel) as the training data, and the remainders for testing. Training data was 246.5 hours from more than 1350 speakers. The test data consisted of two parts, one having 1031 utterances from 4 distinctive speakers in the "LDC" room and the other one having 898 utterances from another 4 speakers in the "HRM" room, respectively.

We tested all the thirteen microphone streams on the typical DNN-HMM system trained on MFCC features, with 11 frames stacking (± 5), shown in Table 1. Except for microphone 2, whose acoustic scene was matched with the training, we derived two test sets for the stream attention task (trained on clean, tested on various conditions). For the "LDC" set, we had twelve streams working in normal status. For the "HRM" set, ten streams worked well for ASR; however, the other two failed (Mic 3&11). This phenomenon happens quite often in real environments, as microphones might be out of charge suddenly or affected by strong echo, noise or reverberation. The system should be robust in case of such microphone failures.

Table 1: *WERs(%) of each distributed microphone stream in the Mixer-6 test sets. The DNN classifier was trained on the recordings from Mic 2, which was not used for testing.*

Mic. ID	LDC room	HRM room
Mic 1	23.5	26.7
Mic 3	26.4	97.6
Mic 4	10.6	**8.2**
Mic 5	12.7	12.6
Mic 6	**9.9**	8.4
Mic 7	15.0	15.1
Mic 8	13.7	12.3
Mic 9	22.6	18.1
Mic 10	11.0	13.2
Mic 11	10.3	75.6
Mic 12	14.3	12.5
Mic 13	19.5	21.6

3.1.2. DIRHA-WSJ dataset (with speaker movement)

The DIRHA-WSJ dataset was collected in a real apartment setting with typical domestic background noise and reverberation. In the configuration, a total of 32 microphones were placed in the living-room (26 microphones) and in the kitchen (6 microphones). The microphone network consists of 2 circular arrays of 6 microphones (located on the ceiling of the living-room and the kitchen), a linear array of 11 sensors (located in the living-room) and 9 microphones distributed on the living-room walls. For the microphone arrays, beamforming outputs can be generated that form extra streams for further comparisons.

A contaminated version of the original WSJ (Wall Street Journal) corpus is used for training, while the test is performed with the DIRHA-WSJ dataset. Both real and simulated data are employed for the test. The real data consisted of 3 Male and 3 Female native US speakers uttering 409 WSJ sentences. We picked out 7 microphones and 1 array distributed in the room for our stream attention purpose. Table 2 shows the baseline WERs for individual microphones, as well as the output from Delay-and-Sum beamforming (LA6 is the central element of the Ceiling Circular Array). During the recording, the speaker was asked to move to a different position and take a different orientation after reading several sentences. To examine the robustness of the proposed system, three representative cases were designed for testing, which were

Case 1: Using 6 individual microphones distributed in the living room (LA6, L1C, L4L, LD07, L3L, L2R) where the WERs of the microphone streams over the test set are comparable as the speaker moved around during recording.

Case 2: Using 5 individual microphones and 1 beamforming output (L1C, L4L, LD07, L3L, L2R, Ceiling Array).

Case 3: Using 6 individual microphones distributed in the living room (LA6, L1C, L4L, LD07, L3L, L2R) and one in the kitchen (KA6), a stream with worst WER.

Table 2: *WERs(%) of the 7 distributed individual microphone streams and 1 beamforming stream in the DIRHA-WSJ test sets, consisting of both simulated and real recordings. Combinations of the streams are used for testing the robustness of the framework.*

Mic. ID	Sim Data	Real Data
LA6	23.7	30.6
L1C	22.5	30.8
L4L	23.5	31.3
LD07	22.4	30.6
L3L	22.6	30.8
L2R	22.8	34.9
KA6	58.6	74.1
Ceiling Circular Array	21.3	26.5

3.2. Baselines and Comparative methods

WER results were compared between the proposed and conventional strategies, like picking out the best stream by detecting the energy [20] or signal-to-noise ratio [21] at the signal level, combining the lattices by doing a union of the lattices from different streams at the lattice level [7], as well as the famous ROVER technique [6]. In addition, we selected a stream with the lowest error rate for each utterance to get the **Utterance Oracle**, and for each test set to get the **Best Stream**, respectively.

M-measure PM was used to select the stream sentence-by-sentence [13][15]. Meanwhile, for the frame-wise fusion, we took inverse entropy [12] and auto-encoder (AE) [16] for performance comparison. A simple **Equal Weights** was performed as the frame-wise baseline [22]. For the auto-encoder hierarchy, we investigated the effect of using different temporal context

Table 3: *WERs(%) comparison of various microphone stream attention approaches on the Mixer-6 and DIRHA-WSJ distributed multi-microphone datasets.* **Group A***: Baselines from cheating experiments;* **Group B***: Conventional strategies performed at the signal, lattice and word level;* **Group C***: Pick out the best stream sentence-by-sentence using M-measure PM in the proposed framework;* **Group D***: Frame-wise stream attention using different PMs.*

Group	Method	Mixer-6 Dataset			DIRHA-WSJ Dataset						
		LDC	HRM	Ave	Case 1		Case 2		Case 3		Ave
					Sim	Real	Sim	Real	Sim	Real	
A	Utterance Oracle	4.1	2.3	3.2	16.1	23.7	15.5	22.6	16.1	23.7	19.6
	Best Stream	9.9	8.2	9.1	22.4	30.6	21.3	26.5	22.4	30.6	25.6
B	Energy	12.7	15.1	13.9	23.3	32.8	22.9	31.2	35.8	50.6	32.8
	Signal-to-Noise Ratio (SNR)	19.5	35.6	27.6	21.5	29.7	21.0	28.8	22.3	30.9	25.7
	Lattice combination	10.7	21.7	16.2	19.9	28.1	19.7	27.1	21.7	31.3	24.6
	ROVER (word level)	7.9	9.8	8.9	19.2	27.8	18.0	26.4	19.5	28.2	23.2
C	M-measure	10.1	9.0	9.6	19.1	27.3	18.1	**26.1**	**19.1**	**27.3**	22.8
D	Equal Weights	9.7	30.0	19.9	19.4	28.3	18.9	27.4	21.4	30.5	24.3
	Inverse entropy	**7.7**	7.7	7.7	19.0	27.6	18.7	27.3	20.7	29.6	23.8
	AE w/o context	8.5	7.0	7.8	18.7	27.6	18.0	26.6	20.0	28.9	23.3
	AE w context [-8, 5]	8.3	6.9	7.6	18.7	27.5	17.9	26.4	19.9	28.6	23.2
	AE w context [-16,12]	8.1	**6.8**	**7.5**	**18.3**	**26.4**	**17.7**	26.3	19.8	28.6	**22.8**
	AE w context [-20,14]	8.4	6.8	7.6	18.9	27.5	18.1	26.2	19.8	28.7	23.2

sizes on WER. The auto-encoders for the Mixer-6 and DIRHA-WSJ dataset were trained with 6 and 5 layers (a 24-unit bottleneck layer in the middle) respectively, and each layer consisted of 512 ReLU units. The context was introduced via a TDNN architecture with different temporal resolutions at each layer.

3.3. Results

Table 3 shows the WER results using various comparative techniques. As shown in Group "A", the **Utterance Oracle** gives the potential best WERs, suggesting that the recognition performance could be largely improved if we do the optimal microphone selection.

In Group "B", we observe that conventional signal level measures, such as energy and SNR, are not reliable. Lattice combination outperforms **Best Stream** in some cases of the DIRHA-WSJ dataset. For the Mixer-6 dataset, Lattice combination performs worse, especially on the "HRM" test set. It takes the risk of involving the bad streams. However, ROVER can provide a stable improvement on both datasets.

Our approach carried out on the DNN posteriors shows a superior performance, which is delivered by Group "C" (sentence based M-measure PM). However, in some applications, the acoustic situation may change dynamically and solutions that require such longer signal spans (over 800ms for the M-measure) for making the stream selection may not be appropriate. For instance, in the Mixer-6 ASR experiments, some sentences are quite short, which cannot provide enough frames with long time span for calculating the M-measure.

In Group "D", we provide the WER results of frame-wise attention using different PMs. In the Mixer-6 dataset, when applying equal weights to the 12 microphone streams, a better WER (9.7%) is achieved on the "LDC" test set. However, performance on "HRM" test set with two of the streams in bad condition gets much worse (30.0%). The same trend can also be observed in the DIRHA-WSJ dataset. The average WER over the cases (24.3%) still shows a better performance than **Best Stream** (25.6%). In contrast, the inverse entropy approach achieves a substantial WER improvement compared to the **Best Stream** one in both datasets. In the Mixer-6 dataset, the relative improvement of the "HRM" set (6.1%) is not as much as that of the "LDC" set (22.2%). This might be due to that the posteriors from pure noises can also derive a low entropy, which misleads the attention scheme. This phenomenon does not occur when the auto-encoder based attention vector is applied. We find that the improvements are consistent in both Mixer-6 test sets. For the DIRHA-WSJ dataset, the auto-encoder approach also gives more robust recognition results than inverse entropy. However, in some cases, especially in the Real test sets, M-measure outperforms the auto-encoder (without context training) approach, suggesting that looking at temporal dynamics of posteriors would be helpful. As we enlarge the context window for training, it is evident that the auto-encoder approach is able to decrease the WER on both datasets. On average, the best performance (7.5% and 22.8%) is achieved when the context window is [-16,12], implying that $\sim$300 ms is enough to complete the stream attention scheme, which is more robust and efficient than M-measure, as well as the ROVER technique.

4. Conclusion

In this work, we aimed at improving the far-field ASR performance using distributed microphones and arrays. A stream attention framework was designed to generate more reliable HMM state posterior probabilities, given a classifier (Improvement of classifier, like classifier adaptation using test data, is not the point in our approach). The attention scheme was achieved by assigning to each stream a discriminative confidence, which is derived via measuring the ASR performance in an unsupervised way. The framework can be flexibly applied to the scenarios with both individual microphones and beamformers.

The far-field ASR experiments revealed that the framework showed a substantial capability to improve the ASR performance compared to the signal, lattice and word level approaches. Among the PM techniques, the extended frame-wise auto-encoder trained with temporal context showed a more robust ability to resist perturbations such as microphone failure and speaker movement. In general, the framework coupling with PMs is able to take advantage of all available microphones in the space in parallel, while decoding for the best path only once, which is computationally affordable and has relatively low latency in the real-time ASR.

5. References

[1] M. Wolf and C. Nadeu, "Channel selection measures for multi-microphone speech recognition," *Speech Communication*, vol. 57, pp. 170–180, 2014.

[2] C. Zhang, D. Florêncio, D. E. Ba, and Z. Zhang, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 538–548, 2008.

[3] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[4] K. Kumatani, J. McDonough, J. F. Lehman, and B. Raj, "Channel selection based on multichannel cross-correlation coefficients for distant speech recognition," in *Proceedings of 2011 Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*. IEEE, 2011, pp. 1–6.

[5] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *Proceedings of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 271–275.

[6] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 1997, pp. 347–354.

[7] H. Xu, D. Povey, L. Mangu, and J. Zhu, "An improved consensus-like method for minimum bayes risk decoding and lattice combination," in *Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 4938–4941.

[8] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[10] S. Kim and I. Lane, "Recurrent models for auditory attention in multi-microphone distance speech recognition," in *Proceedings of Seventeenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 3838–3842.

[11] S. Okawa, E. Bocchieri, and A. Potamianos, "Multi-band speech recognition in noisy environments," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2. IEEE, 1998, pp. 641–644.

[12] H. Misra, H. Bourlard, and V. Tyagi, "Entropy-based multi-stream combination," IDIAP, Tech. Rep., 2002.

[13] H. Hermansky, E. Variani, and V. Peddinti, "Mean temporal distance: Predicting asr error from temporal properties of speech signal," in *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7423–7426.

[14] B. T. Meyer, S. H. Mallidi, A. M. C. Martínez, G. Payá-Vayá, H. Kayser, and H. Hermansky, "Performance monitoring for automatic speech recognition in noisy multi-channel environments," in *Proceedings of Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 50–56.

[15] S. H. Mallidi, T. Ogawa, and H. Hermansky, "Uncertainty estimation of dnn classifiers," in *Proceedings of 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 283–288.

[16] S. H. R. Mallidi, T. Ogawa, K. Veselỳ, P. S. Nidadavolu, and H. Hermansky, "Autoencoder based multi-stream combination for noise robust speech recognition." in *Proceedings of Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015, pp. 3551–3555.

[17] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[18] L. Brandschain, D. Graff, C. Cieri, K. Walker, C. Caruso, and A. Neely, "The mixer 6 corpus: Resources for crosschannel and text independent speaker recognition," in *Proc. of LREC*, 2010.

[19] M. Ravanelli, P. Svaizer, and M. Omologo, "Realistic multi-microphone data simulation for distant speech recognition," *arXiv preprint arXiv:1711.09470*, 2017.

[20] M. Wolf and C. Nadeu, "On the potential of channel selection for recognition of reverberated speech with multiple microphones," in *Proceedings of Eleventh Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

[21] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[22] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.