Combining Deep Neural Networks and Beamforming for Real-Time Multi-Channel Speech Enhancement using a Wireless Acoustic Sensor Network

Ceolini, Enea ; Liu, Shih-Chii

DOI: https://doi.org/10.1109/mlsp.2019.8918787

Accepted for publication at 2019 IEEE International Workshop on
Machine Learning for Signal Processing

COMBINING DEEP NEURAL NETWORKS AND BEAMFORMING FOR REAL-TIME MULTI-CHANNEL SPEECH ENHANCEMENT USING A WIRELESS ACOUSTIC SENSOR NETWORK

Enea Ceolini, Shih-Chii Liu

Institute of Neuroinformatics
University of Zurich and ETH Zurich
Zurich, Switzerland
Email: {enea.ceolini,shih}@ini.uzh.ch

ABSTRACT

This work presents a multi-channel speech enhancement algorithm using a neural network combined with beamforming deployed real-time on a wireless acoustic sensor network (WASN) of distributed microphones. We combine spectral mask estimation via a deep neural network together with spatial filtering to obtain a robust speech enhancement system even in difficult real-world scenarios (e.g. speech in noise, reverberant environments). Although the model is trained on simulated data, it performs comparably well on real-world tasks relative to an ideal oracle beamformer. We show that the model can be deployed on a WASN platform that allows for remote placement of microphones and on-board computing. We consider models with a small parameter count and low computational complexity. It achieves signal-to-distortion ratio (SDR) improvements of up to 10 dB in a real-world scenario and runs real-time on-board the WASN, with a latency in the order of hundreds of milliseconds.

Index Terms— speech enhancement, beamforming, deep neural networks, wireless acoustic sensor networks

1. INTRODUCTION

In recent years, the number of speech enhancement algorithms has grown dramatically [1, 2, 3]. These algorithms are particularly useful for two major applications, that is, automatic speech recognition (ASR) systems [4, 5] and hearing aid [6, 7, 8] technology. Today, ASR systems are deployed in many daily-life applications that require robustness in difficult noise conditions. The same robustness is required by hearing aid technology which sees its market expanding due to hearing loss becoming more common with increased life expectancy [9].

A lot of progress has been made in the field of speech enhancement, nevertheless many of the developed algorithms do not yet take into account the real-world constraints faced by operating the hardware devices that run these algorithms. In particular, these portable hardware devices will have limited memory, computational power, and battery power. Many state-of-art algorithms are based on deep neural network (DNN) models with millions of parameters and high computational costs that render them not usable in embedded devices such as mobile phones or hearing aids [7, 1]. Moreover, most of these algorithms are evaluated on simulated datasets where ground truth of clean speech is available but lack testing in real-world scenarios with real reverberations and non-ideal microphones. Consequently it is hard to say if those algorithms will be robust in real-world conditions [10].

Another trend in deep learning used for speech enhancement is moving past the use of a single channel. Recently, the speech enhancement community has obtained promising results with algorithms that use multiple channels. Using multiple channels can guarantee better results in terms of both SDR and speech intelligibility. In particular using spatial information can lead to better cancellation of point noise sources and diffuse noise [11, 12, 13, 14]. From a technological point of view, the advent of Internet of Things (IoT) and wearable devices brings the possibility of having an ad-hoc microphone array with large inter-microphone spacing and microphones connected together in a wireless acoustic sensor network (WASN). Beamforming algorithms that use the outputs of these ad-hoc arrays can guarantee better SDR then using closely spaced linear or circular arrays [15].

The goal of this work is to design a speech enhancement model that uses an ad-hoc array of randomly placed microphones that is both robust in simulated and real-world scenarios. The model has to be cheap enough to be run on an embedded device, particularly one of the nodes of the WASN platform, under constraints of low computational power and low latency for real-time applications.

The model will be trained on simulated scenarios of speech in noise with the presence of reverberations and evaluated on both the simulated data and real-world data. The real-world data is collected using a readily available multi-microphone distributed platform [16], a scalable WASN that allows for an arbitrary number of microphones to be added to the network. In addition, this platform allows multi-channel models to be interfaced in real-time with all the microphones in the network using its on-board embedded computer. This works shows how a small, but powerful model for speech enhancement can be used in a real-world scenario in real-time.

After a review of related research in Section 2, the remainder of the paper is structured as follows: Section 3 describes the model design and architecture together with the description of the multi-microphone WASN platform. Section 4 describes the experimental evaluation. Finally Section 5 concludes the paper.

2. RELATION TO PRIOR WORK

This work joins current research in speech enhancement algorithms with WASN consisting of low-resource distributed nodes. The considered algorithm in this work uses mask estimation via a neural network together with beamforming [17, 18]. This approach differs

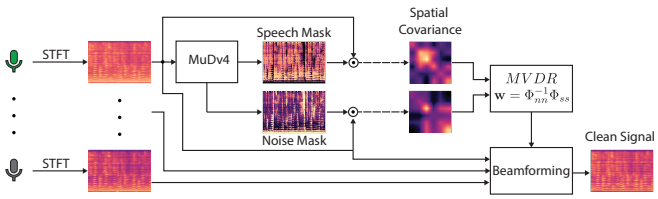

Fig. 1: Architecture of the proposed speech enhancement system. In green is the reference microphone. MuDv4 is the name of the model described in Section 3.2. The sample is speech in $-5\,\mathrm{dB}$ noise.

from approaches that explicitly use spatial features in the model [3, 5]. Our work follows mainly the formulation described in [17], but we use a convolutional neural network (CNN) for the mask estimation. The CNN will be compared to the recurrent (LSTM/BLSTM) and feed forward (FF) models described in [17] in terms of quality of mask estimation and overall run-time. The WASN hardware platform used in this work (WHISPER) was previously used for classical beamforming with an ad-hoc microphone array [19].

In this work, we use the WASN platform to deploy neural networks for speech enhancement running in real-time and in a real-world task. To the best of our knowledge, it is the first time a deep learning based speech enhancement algorithm is deployed in the real world on a WASN. Previous work has been done in developing speech enhancement models usable in real-time with low algorithmic latency, but they are mainly theoretical, meaning without any deployment [7] or they use very simple models that do not achieve outstanding performance [20, 21].

3. METHODS

In this section we describe the signal model, the network architecture used to estimate the speech and noise spectral masks and the WASN system on which the model is deployed for real-time operation.

3.1. Signal model

We consider the following signal model which describes a single sound source within noise. We assume M microphones in the WASN. The signal recorded by channel m can be described as

$$y_m(t) = s(t) * r_m(t) + k_m(t) \qquad i \in 0, ..., M \tag{1}$$

where $s(t)$ is the original clean speech, $r_m(t)$ is the reverberant room impulse response, $k_m(t)$ is the noise at the microphone and $y_m(t)$ is the signal recorded by the microphone. We consider the problem of finding a linear filter that is able to recover the reverberant speech, first term in Eq. 1, and with the least amount of distortion. We assume that in our scenarios the narrowband approximation of the short-time Fourier transform (STFT) holds [15], therefore we analyze this problem in the frequency domain leading to the corresponding form of Eq. 1:

$$Y_m(\omega, n) = S(\omega, n)R_m(\omega, n) + K_m(\omega, n) \tag{2}$$

where (ω, n) represents the time-frequency bin for frequency ω and time n. Traditionally, this problem is optimally solved using spatial filtering, also known as beamforming. This technique leads to the optimal reduction of the noise because of spatial coherence. One popular method is the minimum variance distortionless response (MVDR) beamformer. The MVDR filters are obtained by imposing a constraint of maximally reducing the signal variance, therefore the noise power, without introducing any speech distortion. Traditionally, MVDR weights are obtained by solving the following optimization problem

$$\mathbf{w}_{MVDR} = \arg\min_{\mathbf{w}} \mathbf{w}^H \Phi_{\mathbf{yy}} \mathbf{w} \quad s.t. \quad \mathbf{q}^H \mathbf{w} = 1 \tag{3}$$

where $\mathbf{w}$ are the weights of the filters in the frequency domain, $\Phi_{\mathbf{yy}} = \sum_{n=1}^{N} \mathbf{Y}(n)\mathbf{Y}(n)^H$ is the spatial covariance matrix of the noisy signal, $\mathbf{q}$ is the steering vector and H is the Hermitian transpose operation. This optimization problem yields the solution

$$\mathbf{w}_{MVDR} = \frac{\Phi_{\mathbf{u}}^{-1}\mathbf{q}}{\mathbf{q}^H \Phi_{\mathbf{u}}^{-1}\mathbf{q}} \tag{4}$$

The limitation of this approach, is that it still needs knowledge of the steering vector $\mathbf{q}$, and this vector is usually hard to obtain in practice for an ad-hoc array [19]. To avoid estimating the steering vector, we use an alternative formulation [17] to obtain the MVDR weights [22]. The idea is the same as presented in [17]. The weights are now obtained by using

$$\mathbf{w}_{MVDR} = \Phi_{nn}^{-1}\Phi_{ss} \tag{5}$$

where Φ_{nn} and Φ_{ss} are the noise and speech spatial covariances respectively. In order to compute these spatial covariances, we use the neural network to estimate a spectral mask that roughly separates speech and noise from the mixture. Once we obtain the respective masks M_s and M_n for speech and noise, we calculate the spatial covariances as shown in [18]:

$$\mathbf{\Phi_{vv}} = \sum_{n=1}^{N} M_v \mathbf{Y}(n)\mathbf{Y}(n)^H \tag{6}$$

for $\mathbf{vv} \in \{\mathbf{ss}, \mathbf{nn}\}$ and $v \in \{s, n\}$ and then compute the beamforming weights following Eq. 5.

3.2. Network architecture

As described in Section 3.1, we use a neural network to estimate a time-frequency spectral mask for separating speech from noise. This method works well when we can assume that the noise and speech statistics differ substantially, for e.g. when there is only one speech source which is also the target and the noise is diffuse and relatively stationary.

The input to the network is the mixture spectrogram $\mathbf{Y}_{ref} \in \mathbb{C}^{F \times N}$ of a microphone chosen arbitrarily as the reference microphone; and where F is the number of frequency bins defined by the length of the STFT window and N is the number of time frames in the considered sample. Since the network can only process real numbers, the spectrogram is split in the real and imaginary parts and the two matrices are concatenated along a new axis [23]. The input to the network is now a 3D tensor $\mathbf{I} \in \mathbb{R}^{2 \times F \times N}$. Along with the architectures described in [17], we proposed a new architecture, based on a CNN to estimate the mask.

We use different neural network architectures to estimate the mask. In the case of the convolution neural network, we first apply apply a 2D convolution with kernel size $(1, 1)$, also called a fully connected layer, to transform the number of channels in tensor $\mathbf{I}$ from 2 to 16. Then we apply S stacks of L convolutions, each with 16 channels, kernel sizes of $(3, 3)$ and a dilation factor 2^l where l is the index of the layer in the stack [24]. The convolution outputs are zero padded to keep the matrix dimensions the same as the input dimensions. Pooling is not used. After every convolution we apply

Model	Parameters	Causal	Span (ms)	Latency (ms)	**Real-world data**		**Simulated Data**	
					SDR	STOI	SDR	STOI
C_512_6	30K	yes	3950	15	5.1 ± 2.2	0.63 ± 0.16	11.6 ± 3.8	0.87 ± 0.13
C_512_4	20K	yes	950	15	4.9 ± 2.2	0.62 ± 0.17	11.1 ± 3.9	0.86 ± 0.14
NC_512_6	30K	no	3950	1975	5.6 ± 2.1	0.65 ± 0.15	12.2 ± 3.6	0.89 ± 0.12
NC_512_4	20K	no	950	1975	5.3 ± 2.2	0.63 ± 0.17	11.7 ± 3.7	0.87 ± 0.13
FF [17]	500K	yes	15	15	3.5 ± 2.2	0.58 ± 0.19	9.9 ± 3.9	0.83 ± 0.17
BLSTM [17]	2M	no	*	full sequence	5.7 ± 2.1	0.66 ± 0.17	12.3 ± 3.4	0.87 ± 0.12
LSTM	2M	yes	*	15	5.2 ± 2.0	0.63 ± 0.17	11.5 ± 3.3	0.86 ± 0.15
MB_MVDR	-	-	-	-	6.5 ± 1.9	0.72 ± 0.12	12.6 ± 1.7	0.91 ± 0.10

Table 1: Results of the models and of a reference mask-based beamformer on simulated and real-world data using 6 microphones. Reported are mean and standard deviation values for SDR and STOI. The mask-based beamformer (MB_MVDR) has been obtained using an IRM. The numbers in the *parameters* column include the total number of trainable parameters in the model. The *span*, also known as receptive field, and the *latency* are calculated using a sampling frequency of 8 kHz, a STFT frame size of 512 and a frame step of 125 samples. The * in the span for LSTM-based models indicates that the true span cannot be calculated for these models.

a ReLU non-linearity and a layer normalization [25]. In addition, the input to the next layer is summed with identity residual connections to make the training faster and more stable [26]. Finally, a fully connected layer brings the channels down from 8 to 1, leading to a magnitude spectral mask $M \in \mathbb{R}^{F \times N}$.

In order to reduce the number of parameters we only estimate M_{ss} and assume that $M_{nn} = 1 - M_{ss}$ as in an ideal ratio mask (IRM) [27]. As the next step, we calculate the spatial covariances as shown in Eq. 6 and the MVDR weights as in Eq. 5. Finally, we apply the weights to the mixture spectrogram $\mathbf{Y}$ and obtain the estimated cleaned speech. Details on the training procedure will be given later in Section 4. Note that since the network estimates only one mask from the reference microphone, the enhancement model is inherently applicable to an array with an arbitrary number of microphones. As shown in [17], we could have employed a more complex mask estimation scheme that makes use of all microphones. Nevertheless, the limited boost in performance is not worth increasing the computational complexity of the model and it will badly affect the speed performance on a low-resource embedded device.

3.3. Wireless acoustic sensor network platform

We use the WHISPER WASN platform to deploy the models in the real world. Each module of the platform holds up to 4 microphones and has both high and low level computation capabilities powered respectively by an FPGA and an embedded computer namely a Raspberry-Pi 3 Model B+, featuring a Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC running at 1.4GHz and a 1GB LPDDR2 SDRAM.

All modules can be wirelessly synchronized in order to maintain synchronous sampling across all microphones even on different modules. The synchronization precision is in the order of nanoseconds. This value is enough to deliver sample-level synchronization even at high sampling rates. The modular design allows WHISPER to be scaled to arbitrary number of modules and therefore microphones that can be deployed in such a way to construct an ad-hoc WASN. Large spacings (in the order of meters) between microphones allow better results in terms of SDR [15].

We first collected a real-world dataset using WHISPER of a single talker at different SNR levels and in the presence of reverberation (see Section 4.2). We used this dataset to test our model and to compare the performance of the model with classical oracle beamforming on a real-world task. The model is also deployed onto the WASN platform so that we can evaluate its real-time performance in a real-world scenario.

4. EXPERIMENTAL EVALUATION

The evaluation is done in three different test scenarios. The first test is carried out using the simulated data also used for training. The second test is done offline on the real-world dataset recorded using the multi-channel platform. The third test is the deployment of the models in a real-world scenario by using WHISPER. For the offline test, we show the results in terms of signal quality (SDR) and speech intelligibility (short-time objective intelligibility (STOI)). For the real-time scenario, we report on system latency, computational load and memory usage for the model deployed on the platform. For all three cases, the signals are sampled at 8 kHz and the STFT is computed using a frame length of 64 ms and frame step of 15 ms.

The newly introduced model described in Section 3.2 are compared against the two models introduced in [17] for mask estimation, namely a bidirectional recurrent neural network (RNN) with long-short term memory (LSTM) units and a feed forward network. Moreover, we compare against a version of the RNN in [17] that uses only unidirectional units in order to have a causal network. This network is similar to the bidirectional LSTM network but has double the number units, this is done to match the number of operations to compute this network to the number of operations needed to compute the smallest CNN. Finally, all models are compared against a mask-based MVDR oracle beamformer.

4.1. Simulated data

The models were trained on a dataset of simulated data. The dataset is based on the WSJ0-2mix dataset [28] commonly used for source separation and speech enhancement tasks. We generated simulated scenarios of a single speaker in a reverberant room and in the presence of diffuse noise. The noise is taken from NoiseX-92, [29] a widely known dataset containing various types of noise recordings such as factory and babble noise. The noise is mixed with the speech at SNR levels in the range of $(-7.5, 2.5)$ dB. For each sample, we randomly selected a room size of $\mathcal{U}(3,6) \times \mathcal{U}(3,6) \times \mathcal{U}(3,4)$ meters, a reverberation time $T_{60} = \mathcal{U}(0.2, 0.8)$ seconds, a sample from the noise dataset, the positions of 6 microphones and the position of

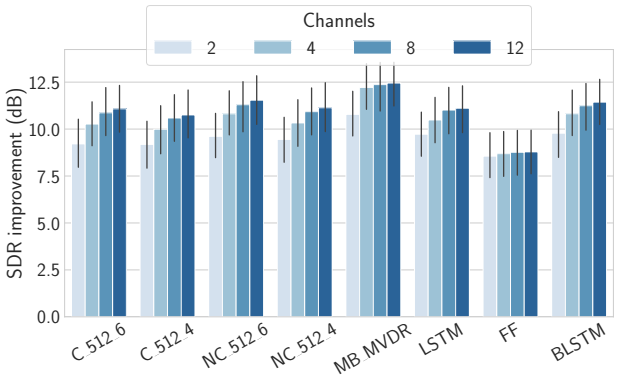

Fig. 2: SDR improvement for the different models and the mask-based beamformer on the real-world data as a function of the number of microphones used. Results are averaged across the 3 SNR levels.

Model	**R** (ms)	**T** (ms)	**M** (MiB)	**F** (Hz)
C_512_4	315 ± 7	330 ± 7	66.14	3.17
C_512_6	1773 ± 20	1788 ± 20	80.13	0.6
NC_512_4	321 ± 8	2296 ± 8	66.14	3.11
NC_512_6	1826 ± 25	3801 ± 25	80.13	0.5
FF	107 ± 4	122 ± 4	30.18	9.34
LSTM	317 ± 9	332 ± 9	70.41	3.15

Table 2: Results for real-time operation of the models on the platform. (**R**)untime is the time taken to process a frame of length matching the receptive field of the model. (**T**)otal latency includes algorithmic latency (Table 1) plus (**R**)untime. (M)emory is overall dynamic memory usage not storage. (**F**)rame rate refers to the maximum rate at which the new weights can be calculated.

the source. We impose no constraints on the minimum or maximum distance between each microphone and the source.

The model is trained to estimate a speech mask as described in Section 3.2, nevertheless the optimization of the network is done in the time domain. Since both the calculation of the weights (Eq. 5) and the STFT are differentiable operations, we can invert the STFT signal in the time domain and train the model end-to-end using the scale-invariant SDR as the target [30]. The SDR is computed from the estimated and ground-truth reverberant speech. The model is trained for 100 epochs. In each epoch, the training data consists of $20,000$ samples, each of 4 seconds in duration. We use the Adam optimizer, a learning rate of $1e^{-4}$, and early stopping for regularization with a patience of 2 epochs.

The results are reported in Table 1 for 4 model variants all using 6 microphones. They are compared against an oracle MVDR mask-based beamformer that uses the IRM to calculate the spatial covariances of speech and noise. All CNN models have $S = 2$ and either $L = 6$ or $L = 4$ (last digit in model name). As we can see, the non-causal model with the longest receptive field (span) performs the best and has comparable results to the oracle beamformer and to the BLSTM model previously proposed. The simplest model is causal, has low latency and small receptive field. It only has a drop of 1 dB in SDR and a negligible drop in speech intelligibility with respect to the bigger models. On the other hand, it has 1.5 times fewer operations thus making it more affordable in terms of runtime and memory usage. Moreover, even though the small causal model has worst results than its non-causal counterpart, it is more suitable for real-time operations given the short algorithmic latency.

4.2. Real-world data

We collected a dataset using the multi-channel audio platform. This dataset [1] consists of samples of a talker in the presence of babble noise at 3 different SNR levels namely $[0, 5, 10]$ dB. The position of the talker can be in one out of 4 possible locations. The dataset was recorded in a room which has a reverberation time of $T_{60} = 0.25$ s and is of size $5\,\mathrm{m} \times 6\,\mathrm{m} \times 4\,\mathrm{m}$. We used a set of 16 distributed microphones of WHISPER, i.e. 4 wirelessly synchronized modules. The microphones were placed once in an arbitrary configuration. The dataset has 30 samples for each SNR level. Each sample consists of 15 sec of the speaker talking alone, followed by 15 sec of only noise and another 15 sec of noisy speech. The first 15 sec will be used as ground truth to obtain the SDR and STOI objectives that are used to calculate the masks for the oracle beamformer.

Note that the models are not retrained using this real-world data and the results are simply obtained by applying the models on this data. Moreover, the data has not been preprocessed in any way. During evaluation, the model received the raw sampled data from the microphones.

Table 1 shows the results using ad-hoc combinations of 6 microphones in order to have results comparable with the the simulated data. For each of the samples in the real-world test dataset, we randomly select 3 subsets of 6 microphones and evaluate the performance.

As pointed out in [17], these results should be taken with a grain of salt, given the lack of real ground truth as one has in the case of simulated data. Nevertheless, we can compare the models with respect to the oracle beamformer which sets an upper bound to the best possible performance. In accord with the simulated data results, the best model is the causal one with the largest receptive field. In this case, the model performs well even compared to the oracle beamformer which leads by less then 1 dB in SDR. Again, the smallest model is the worst, but has a performance which is still comparable and competitive with respect to the oracle beamformer.

Figure 2 gives a closer look at the performance of the models. Here, we can see the SDR improvement instead of the raw SDR given in Table 1. The SDR improvement is calculated as the difference between the SDR of the enhanced speech and the SDR of the unprocessed speech. The results are averaged over different combinations of the microphones from the original sixteen microphones, depending on the number of microphones. In particular we randomly selected 3 subsets of microphones for each ad-hoc array size. This measure gives a better idea of the power of the models in cleaning the speech, given the lack of ground truth. As we can see, the smallest model using 12 microphones has only 1 dB drop in SDR improvement with respect to the oracle beamformer, but still delivers an overall 10 dB SDR improvement. Moreover, all the models have a significant positive effect with increasing number of microphones used. The power of the algorithm comes from the robustness of the individual components, namely the mask estimation using the deep network and the spatial filtering.

[1] https://github.com/SensorsAudioINI/WHISPER_SET_1

4.3. Real-time operation

We deployed the four convolutional models, the unidirectional LSTM model, and the feed-forward model onto WHISPER using 2 modules (therefore 8 microphones) and measured the real-time performance which is reported in Table 2. The bidirectional LSTM model is not considered because it needs a full sequence to estimate the mask and is not suitable for real-time applications where low-latency is required. The network operations are all carried out in floating point and the model parameters are not quantized. No specific optimization is employed on the CPU but all models are implemented using the ONNX runtime library [2].

The results show that the feed-forward model has the fastest runtime. Nevertheless as shown from the real-data evaluation, this model does not yield satisfactory results in terms of SDR improvement so it might not be the best choice for real-world deployment. In order to get better mask estimates, the RNN and CNN models are preferred. As demonstrated in Table 2, the runtime for these models are similar. Even though they have a significant difference in the number of parameters, the number of operations to compute the models are comparable thus giving a similar runtime. In general, given that both the RNN and CNN give similar performances in terms of the mask estimation quality and runtime, one can choose which network to deploy dependent on the computational efficiency (operations/sec/W) and on-chip memory resource of the target hardware.

The runtime of the recurrent model and the small CNN models is around $300\,\mathrm{ms}$ which gives a frame rate of $3\,\mathrm{Hz}$ for refreshing the weights. For a mask-based algorithm that directly enhances the signal, the requirement of a low latency in the system means that the processing of one frame has to be completed before the next frame arrives. In our case we do not have this constraint since we do not use the mask to clean the signal, but rather to estimate spatial covariances. Once the beamforming weights are computed, they can be applied with a minimum latency of $15\,\mathrm{ms}$ which is set by the STFT frame length chosen in this work while new weights can continually be computed in parallel.

If the sources are spatially stationary, using either the small or the large CNN models does not make a difference because even with the different delays in estimated new beamforming weights, the values would remain the same, so the beamforming performance would not be affected. If the auditory scene changes, e.g. the speaker moves, the new estimated weights would be available only after 300ms for the small and causal model while this delay would be at least $1.79\,\mathrm{s}$ for the larger and non-causal model. Therefore, using a bigger model would lead to inaccurate results. Naturally, if one knows that the auditory scene changes very slowly, one could choose to deploy a bigger and better model.

5. CONCLUSION

This work describes how a system that combines a deep neural network for mask estimation together with beamforming can deliver competitive speech quality improvement both on simulated and real-world data. We compare the performances of different deep network architectures for the mask estimation. Moreover, we showed how a small version of the model can be deployed on a WASN platform of distributed microphones to deliver real-time performance of speech enhancement; and that even a model with only a few thousand parameters can deliver high SDR improvement in a real-world scenario and in real-time with low latency. Although single-channel solutions are also appropriate for the single talker scenario in this study, our intention is to expand the system testing towards multi-talker scenarios in the real world.

[2] `https://github.com/microsoft/onnxruntime`

6. ACKNOWLEDGEMENTS

This work was partially founded by the Swiss National Science Foundation grant agreement No. 200021_172553. The authors would like to thank Yi Luo and Nima Mesgarani for the enlightening and fruitful discussions.

7. REFERENCES

[1] S. Nie, S. Liang, B. Liu, Y. Zhang, W. Liu, and J. Tao, "Deep noise tracking network: A hybrid signal processing/deep learning approach to speech enhancement," in *Interspeech 2018*, 2018.

[2] H. Zhao, S. Zarar, I. Tashev, and C. Lee, "Convolutional-recurrent neural networks for speech enhancement," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2401–2405.

[3] Z. Q. Wang and D. L. Wang, "All-neural multi-channel speech enhancement," in *Interspeech 2018*, 2018.

[4] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and Li H., "On time-frequency mask estimation for mvdr beamforming with application in robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 3246–3250.

[5] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, Li B., E. Variani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, *Raw Multichannel Processing Using Deep Neural Networks*, chapter 4, pp. 105–133, Springer, 2017.

[6] G. Naithani, T. Barker, G. Parascandolo, L. Bramslow, N. H. Pontoppidan, and T. Virtanen, "Low latency sound source separation using convolutional recurrent neural networks," *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 71–75, 2017.

[7] K. Tan and D. L. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Interspeech*, 2018.

[8] J. O'Sullivan, Z. Chen, J. Herrero, G. Mckhann, S. A. Sheth, A. D. Mehta, and N. Mesgarani, "Neural decoding of attentional selection in multi-speaker environments without access to clean sources.," *Journal of Neural Engineering*, vol. 14, no. 5, pp. 056001, 2017.

[9] H. J. Hoffman, R. A. Dobie, K. G. Losonczy, C. L. Themann, and G. A. Flamme, "Declining prevalence of hearing loss in us adults aged 20 to 69 years," *JAMA OtolaryngologyHead and Neck Surgery*, vol. 143, no. 3, pp. 274–285, 2017.

[10] L. Girin, S. Gannot, and X. Li, "Chapter 3 - audio source separation into the wild," in *Multimodal Behavior Analysis in the Wild*, Xavier Alameda-Pineda, Elisa Ricci, and Nicu Sebe, Eds., Computer Vision and Pattern Recognition, pp. 53 – 78. Academic Press, 2019.

[11] X. Zhang and D. L. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM*

transactions on audio, speech, and language processing, vol. 25, pp. 1075–1084, May 2017.

[12] Y. Jiang, D. L. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, 2014.

[13] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, April 2017.

[14] P. Pertilä and J. Nikunen, "Distant speech separation using predicted timefrequency masks from spatial features," *Speech Communication*, vol. 68, no. C, pp. 97–106, April 2015.

[15] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*, John Wiley and Sons, Hoboken, NJ, 08 2018.

[16] I. Kiselev, E. Ceolini, A. d. Cheveigne D. Wong and, and S. Liu, "Whisper: Wirelessly synchronized distributed audio sensor platform," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, Oct 2017, pp. 35–43.

[17] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Interspeech*, 2016.

[18] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 196–200.

[19] E. Ceolini, J. Anumula, A. E. G. Huber, I. Kiselev, and S.C. Liu, "Speaker activity detection and minimum variance beamforming for source separation," in *Interspeech*, 2018.

[20] Y. Rao, Y. Hao, I. M. S. Panahi, and N. Kehtarnavaz, "Smartphone-based real-time speech enhancement for improving hearing aids speech perception," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5885–5888, 2016.

[21] A. Sehgal A. Bhattacharya and and N. Kehtarnavaz, "Low-latency smartphone app for real-time noise reduction of noisy speech signals," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1280–1284.

[22] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 260–276, 2010.

[23] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation," *ACM Trans. Graph.*, vol. 37, pp. 112:1–112:11, 2018.

[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016.

[25] J. Ba, R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[27] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, Dec 2014.

[28] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35, 2016.

[29] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NoiseX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, July 1993.

[30] Y. Luo and N. Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 696–700.