Interspeech 2018
2-6 September 2018, Hyderabad

Investigations on Data Augmentation and Loss Functions for Deep Learning Based Speech-Background Separation

Hakan Erdogan and Takuya Yoshioka

Microsoft AI and Research, One Microsoft Way, Redmond, WA, USA
{hakan.erdogan,tayoshio}@microsoft.com

Abstract

A successful deep learning-based method for separation of a speech signal from an interfering background audio signal is based on neural network prediction of time-frequency masks which multiply noisy signal's short-time Fourier transform (STFT) to yield the STFT of an enhanced signal. In this paper, we investigate training strategies for mask-prediction-based speech-background separation systems. First, we examine the impact of mixing speech and noise files on the fly during training, which enables models to be trained on virtually infinite amount of data. We also investigate the effect of using a novel signal-to-noise ratio related loss function, instead of mean-squared error which is prone to scaling differences among utterances. We evaluate bi-directional long-short term memory (BLSTM) networks as well as a combination of convolutional and BLSTM (CNN+BLSTM) networks for mask prediction and compare performances of real and complex-valued mask prediction. Data-augmented training combined with a novel loss function yields significant improvements in signal to distortion ratio (SDR) and perceptual evaluation of speech quality (PESQ) as compared to the best published result on CHiME-2 medium vocabulary data set when using a CNN+BLSTM network.
Index Terms: source separation, deep learning, speech denoising, speech enhancement

1. Introduction

Speech enhancement and audio source separation has been interesting research topics for researchers in the audio signal processing area. Recently, deep learning based supervised speech-background [1, 2, 3] and speech-speech separation [4, 5] efforts have intensified and such systems started achieving much better results as compared to their alternatives, be it conventional unsupervised methods, or other alternatives based on supervised or semi-supervised learning, such as approaches using nonnegative matrix factorization [6, 7].

Some earlier studies compared separation methods based on direct prediction of magnitude spectra of individual sources (or one target source) versus mask prediction [8, 9, 10]. The mask prediction approach estimates a multiplier of the mixed spectrogram which would yield one of the sources. While the results reported thus far are mixed, in this paper, we attempt to improve the mask prediction approach in the speech-background separation tasks.

We investigate two aspects of the mask-based speech-background separation approach. The first aspect is the data that we use to train these mask prediction models. Data augmentation has been successfully used in other areas of deep learning, such as image classification [11] and speech recognition [12, 13, 14]. For example, the noise robustness of speech recognition systems can be improved by adding reverberated and noise-corrupted versions of original speech utterances to the training set. Data augmentation for source separation tasks is quite straightforward to come up with. By having a training set which is formed by arbitrarily mixing two sources during training would cover a much larger variety of mixing cases for a neural network to learn from. Then, the question comes how many data points do we need? In the past, the models were usually trained on fixed amounts of mixture data by creating them prior to training. In this paper, we explore the effect of mixing the utterances on the fly during training. This allows us to train the model on virtually infinite quantity of data.

The second aspect is the loss function for model training. We believe we have not yet attained the best loss function for audio source separation that works well in terms of convergence behaviour on the training data as well as generalizability to unseen data. We introduce new loss functions and explore their potential in the speech-background separation problem. The new loss functions do not get affected by scaling of the training utterances which we believe is an important quality.

2. Problem Definition

In speech-background separation, we seek to separate sources from an observed mixed signal $y[k] = s[k] + n[k]$ where $y[k]$ denotes the observed signal, $s[k]$ is the first source which is typically speech, and $n[k]$ is the second source which is typically noise or other interference like music in the background. It has been beneficial to work in the short-time Fourier transform (STFT) domain in which we have the additivity $Y(t, f) = S(t, f) + N(t, f)$ where (t, f) indicates the time-frequency bin of interest and $Y(t, f)$ is the complex-valued discrete Fourier transform of a windowed signal at frame t.

Our goal is to recover $S(t, f)$ and $N(t, f)$ given $Y(t, f)$. In order to overcome the under-determined nature of the problem, we have to make use of some properties of the sources. Deep learning can implicitly leverage the source-related properties by training a source separation model on a lot of clean-mixture pairs in a supervised manner.

3. Mask Prediction

Mask prediction refers to finding a mask, say $M_s(t, f)$ which helps us get an estimate of the source signal's STFT from the mixture signal as $\hat{S}(t, f) = M_s(t, f)Y(t, f)$. We can define ideal masks in different ways and seek to estimate these masks for improved prediction of the sources. We will focus on two types of real-valued masks in this paper. First one is an ideal amplitude mask (IAM), which is defined as $M_s^{\text{IAM}}(t, f) = |S(t, f)|/|Y(t, f)|$. The IAM yields the exact magnitude of the first source when applied. A phase-sensitive filter (PSF) is defined as $M_s^{\text{PSF}}(t, f) = |S(t, f)| \cos(\theta_{sy})/|Y(t, f)|$, which yields the lowest squared error estimate of the source in the complex STFT domain, given the mask is real-valued [15].

10.21437/Interspeech.2018-2441

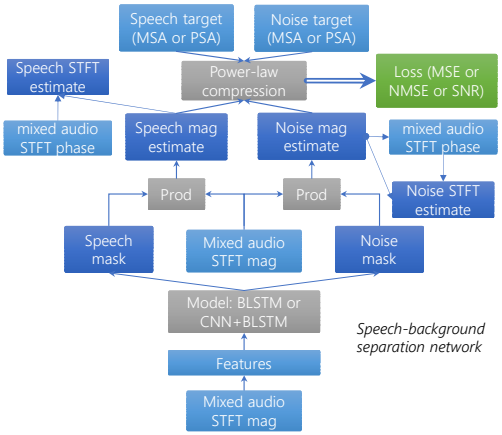

Figure 1: *System architecture.*

Here θ_{sy} is the angle between two complex numbers $Y(t,f)$ and $S(t,f)$. In the context of deep learning based mask prediction, we find ideal ratio mask (IRM) and Wiener-like masks [8] worse than these two alternatives, hence we do not consider them here. We also compare with complex IRM (cIRM) prediction [16, 17] in our experiments.

We use the system architecture shown in Figure 1 for predicting masks for both of the individual sources in the mixture.

4. Loss Functions for Mask Prediction

Although we would like our networks to predict masks for reconstruction of sources, it is not immediately clear what constitutes a good loss function for that purpose. Actually, there are many possible loss functions and many different ones were considered in the literature. While designing or specifying loss functions, we need to be careful about a few things.

1. The loss function should relate to the performance metric we desire to optimize on test data.
2. While having a good convergence behavior, the loss function should avoid over-fitting to the training data, so that it would generalize well on unseen test data.

Initial studies of mask prediction used mask domain mean squared error loss function which can be written as follows:

$$\frac{1}{N}\sum_{u,t,f}(O_u(t,f)-M_u^*(t,f))^2.$$

Here u is the utterance index and $O_u(t,f)$ is the output of the mask prediction network at (t,f) time-frequency bin of utterance u and N is the total number of time-frequency bins for all utterances and M_u^* indicates the ideal mask target which can be IAM or PSF ideal mask.

In [3, 8], we used a signal domain loss function where the error is measured in the domain of predicted signals: $1/N\sum_{u,t,f}(O_u(t,f)|Y_u(t,f)|-M_u^*(t,f)|Y_u(t,f)|)^2$, which is equivalent to a weighted version of the previous mask domain loss where the weight is the energy of the mixed signal at each time-frequency bin:

$$\frac{1}{N}\sum_{u,t,f}|Y_u(t,f)|^2(O_u(t,f)-M_u^*(t,f))^2.$$

This loss function is arguably better than the mask-domain loss since whereas the mask-domain loss function weights each time-frequency bin equally, this one gives more importance to the high energy bins which matter more in calculating signal-to-reconstruction-noise ratios (SNR) which we care about as a metric of performance.

We called these losses magnitude spectrum approximation (MSA) and phase-sensitive spectrum approximation (PSA) loss functions in [8, 15] for prediction of IAM and PSF masks respectively. In the following discussion, we call "targets" for magnitude signal prediction as MSA or PSA targets and use the symbol $S_u^*(t,f) = M_u^*(t,f)|Y_u(t,f)|$ for them, where M^* is the IAM mask for an MSA target and the PSF mask for a PSA target. For predictions of these targets, we use the symbol $\hat{S}_u(t,f) = O_u(t,f)|Y_u(t,f)|$.

4.1. Problem with Signal-domain Losses

One possible problem with the signal domain loss functions, such as MSA and PSA is that they are prone to scaling differences among utterances. So, if an utterance has higher energy overall than another, it is weighted more in the loss function which is not desired. We would like training to generalize to unseen utterances, so we do not want the training to be balanced towards higher energy utterances or even higher energy regions within utterances but learn equally well to separate any possible mixture of speech and background.

One way to mitigate this effect is to use a compression function on the entities to be predicted and the prediction itself before taking the squared error. Hence, instead of using the loss $D(s,\hat{s}) = (s-\hat{s})^2$, we use the following $D(s,\hat{s}) = (f(s)-f(\hat{s}))^2$, where $f(.)$ is a compression function. One possible compression function is the logarithm $f(s) = \log(s)$. When it is used, the loss is equivalent to the squared error in the log-spectrum domain. However, we found that power-law compression is more stable and works better $f(s) = s^\alpha$ where $\alpha \in (0,1]$ is a parameter. Using compression makes large values smaller and small values relatively larger in the loss function and avoids high energy regions from dominating the loss function and making low energy regions/utterances relatively more important. When we use compression, we apply it to both the predicted and the target spectra, making our new MSE loss function: $1/N\sum_{u,t,f}(\hat{S}_u(t,f)^\alpha - S_u^*(t,f)^\alpha)^2$.

Another way to avoid scale differences among utterances is to normalize losses for each utterance with the total energy in the utterance (possibly after compression) which we call the normalized mean-squared error (NMSE) loss function:

$$\frac{1}{\sum_u w_u}\sum_u \frac{w_u}{\sum_{t,f\in u}(S_u^*(t,f)^\alpha)^2}\sum_{t,f\in u}(\hat{S}_u(t,f)^\alpha - S_u^*(t,f)^\alpha)^2.$$

This loss function somewhat takes care of the energy differences between utterances. Here w_u is a weight for each utterance. In this paper, we promote the use of an "SNR" loss function which we define as follows:

$$-SNR = \frac{-10}{\sum_u w_u}\sum_u w_u\left\{\log_{10}\left(\sum_{t,f\in u}(S_u^*(t,f)^\alpha)^2\right) - \log_{10}\left(\sum_{t,f\in u}(\hat{S}_u(t,f)^\alpha - S_u^*(t,f)^\alpha)^2\right)\right\}.$$

This is actually the negative of the SNR in dB of the reconstructed signal with respect to the target magnitude of interest. w_u are weights for each utterance, which can either be all one

or they can be equal to the length of each utterance. Note that, we kept the power-law compression in place, since we found it still to be beneficial even after using the SNR loss function. Power-law compression evens out energy differences within an utterance and the SNR loss function evens out energy differences among utterances. Also note that, both NMSE and negative SNR loss functions are not affected by scaling of utterances, and hence they may be more desirable than the MSE loss. The SNR is calculated in magnitude-STFT domain and this is closely related to the SNR in time domain which we care about as a performance metric.

Even though SNR loss is a good one that matches our test criterion, it may have a drawback for certain training data. Since the value of SNR in dB for an utterance is unbounded, we may have the network focus on "easy" utterances since it can get "cheap" gains by increasing their SNR up to infinity and ignoring all other utterances. This could happen if there is noiseless data in the training set and the network can get infinite SNR by outputting all ones as a mask regardless of the input. To avoid these extreme cases, a possible solution is to compress the SNR loss with a function like $\text{SNR}' = A\tanh(\text{SNR}/A)$ which would limit the output SNR value to be between $-A$ and A and saturate the range. Hence, we used the compressed version of SNR loss in this paper.

4.2. Mel Domain Loss In Early Stages

During training, in early stages, we also make use of a Mel-domain loss function which is obtained by transforming predicted and true spectral magnitudes with a linear Mel-transform. After Mel transformation, we obtain a lower resolution Mel-domain spectra as compared to full-length higher dimensional spectra. We apply power-law compression to the Mel-domain transformed spectra as well. This seems to help during initial epochs of training to guide the network to have better convergence. With the Mel domain loss, we basically replace $D(\boldsymbol{s}, \boldsymbol{s}') = ||(\boldsymbol{s})^\alpha - (\boldsymbol{s}')^\alpha||$ with $D((\boldsymbol{s}, \boldsymbol{s}') = ||(\boldsymbol{M}\boldsymbol{s})^\alpha - (\boldsymbol{M}\boldsymbol{s}')^\alpha||^2$ which means that we process each of the four magnitude spectra in Figure 1 with a linear Mel transform before applying power-law compression.

4.3. Predicting Double Masks

In this paper, we always predict two masks, one for each source, instead of focusing on prediction error of a single source. Our loss functions are a sum of loss functions for each source.

Each mask has an infinite possible range and it is beneficial to limit their ranges. We consider the triangle depicted in Figure 2 and focus on the prediction of IAMs. Instead of predicting double masks directly, we predict their sum and difference. From the relation between sides of a triangle, we know that the sum of the magnitude masks $\sigma = M_s + M_n \geq 1$ and their difference satisfies $-1 \leq \delta = M_s - M_n \leq 1$.

Even if the sum of masks is unlimited, we may limit the sum to be between 1 and 2 (since sum of masks to be larger than two is unlikely) and use a shifted sigmoid output nonlinearity for the network predicting the sum of the masks, $1 + 1/(1 + e^{-x})$. We use a tangent hyperbolic output nonlinearity for predicting the difference of masks which fits its range. From the sum and the difference, we predict the individual masks using $O_s = 0.5(\sigma + \delta)$ and $O_n = 0.5(\sigma - \delta)$.

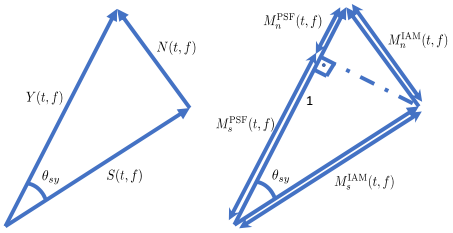

Figure 2: *Double masks illustrated in the complex domain.*

4.4. Post-transformation of Double Masks

At test time, it is beneficial to transform the predicted double masks as follows. For amplitude predicting masks (MSA), we can shrink the masks by calculating their projection in the direction of the mixed signal. From the triangle in Figure 2, these projections can be found with a transformation as follows for MSA: $O'_s = 0.5(1 + O_s^2 - O_n^2)$, and $O'_n = 0.5(1 + O_n^2 - O_s^2)$. For PSA trained masks, we know that the ideal phase-sensitive filters should sum to one, so we can use that constraint at test time to get $O'_s = 0.5(O_s + (1 - O_n)$ and $O'_n = 0.5(O_n + (1 - O_s)$ for the speech and noise masks. After these transformations, we found that MSA trained masks get closer in performance to PSA trained masks.

5. Data Augmented Training

Our version of data augmented training uses speech and noise files that are in the original training data set and does not introduce any additional new data for training. In that sense, it is a "fair" use of training data, since training data includes both mixed signals as well as the clean sources corresponding to them.

However, we form novel mixtures on-the-fly as mentioned earlier. A random speech signal is taken from the training set and it is shifted randomly between $-L/2$ and $L/2$ samples where L is the STFT frame shift in samples. Then another noise utterance is randomly chosen. If the noise utterance is longer in length, a random segment in the noise utterance is added to the speech signal at a random SNR value within the range of SNRs considered in the evaluation set. If the noise signal is shorter, it is doubled in length by repetition until it is longer than the speech signal and a random segment within the repeated noise file is chosen to be added. This way we obtain virtually infinite possible combinations of speech and noise files in the original training set and the training never sees the same mixed signal twice. Each new signal is a novel mixture for the training algorithm. This way of training seems to avoid overfitting to the training data and regularizes training.

6. Experiments and Discussion

We performed speech-background separation experiments on the CHiME-2 medium vocabulary data set [18]. Earlier studies on this data set can be found in [8, 15, 3].

As input to the network, we experimented with 100 or 200 dimensional Mel features. When computing such high-dimensional Mel features, we linearly interpolated the input magnitude spectra to avoid zero Mel-filterbank energies. The STFT parameters were a frame shift of 160 samples, a window size of 480 samples, and a DFT size of 512.

We considered two different neural networks for mask prediction. Our base network is either a feed-forward followed by two layer BLSTM network with 400 hidden nodes or a CNN

+ BLSTM network. The CNN network had two convolutional layers and a 3-D pooling layer with the first one with 15x3 kernels and 32 output channels and the second one with 3x3 kernels and 64 output channels. There is no nonlinearity between two convolutional layers. The pooling layer pools over 3x1x3 blocks, where the first dimension is channels and the last dimension is frequencies. Thus, we never pool in the time dimension since we process whole utterances as 2D images[1]. The pooled CNN output is transformed into a 2D shape and fed into a BLSTM with 300 nodes. Both these base networks have an output dense linear layer with an output length of three times the spectrum size with hyperbolic tangent nonlinearity. The base networks are followed by a feed-forward output layer which has two outputs for double masks prediction (through their sum and difference as mentioned in Section 4.3) or real and imaginary masks for cIRM prediction [16] which we performed for comparison.

We define an epoch as having 1000 utterances (about 2 hours) and train with a variable frame-size minibatch of 5 utterances. For the first 20 epochs, we use a Mel-domain transformed and power-law compressed loss function with 80 Mel dimensions and a coefficient of $\alpha = 1/5$, then for the following 20 epochs, we use 160 Mel dimensions and a $\alpha = 1/3$ in the loss function as described in Section 4.2. Afterwards, we use full spectrum loss and an $\alpha = 1/2$ is used. We fix the learning rate for the initial 100 epochs and decrease it linearly between 100-300 epochs down to a factor of 100 using ADAM [19] optimizer, where we start with a learning rate of 0.002. We also use annealed dropout [20], where we start with an initial dropout rate of 0.5 and reduce it linearly after 50 epochs. We always conclude training after 300 epochs.

6.1. Original and Data-augmented Training

Figure 3 shows the training and validation losses for the models trained on the original training set and with the proposed on-the-fly data augmentation when using a BLSTM network with an SNR loss function. The loss functions shown are the negative signal-to-noise ratios in the complex STFT domain which closely relate to the time-domain SNR. We can see that on-the-fly data augmented training mitigated overfitting to training data, achieving a lower validation loss than the model trained without it.

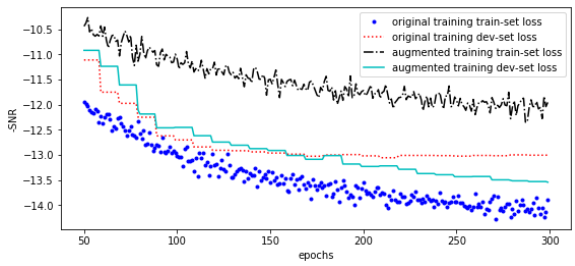

Figure 3: *Negative SNR progress for original and on-the-fly augmented training. Validation (or dev) set evaluated once every 10 epochs.*

Table 1 compares the best published result in the literature on CHiME-2 wsj0 (medium vocabulary) data set with the results we get using our networks with the original training data. The metrics we used are speech to distortion ratio (SDR) and speech to interference ratio (SIR) [21], perceptual evaluation of speech quality (PESQ) [22] and short-time objective intelligibility (STOI) [23]. The results show that we can obtain similar performance to our earlier result in [15] with the networks we consider in this paper using the same original training data and an MSE loss function. In these experiments, we used 100 cuberoot Mel features as input to the network.

In the second part of Table 1, we show the results using on-the-fly data-augmented training and novel loss functions. The best SDR result is obtained with an SNR loss function using a PSA target when using data-augmented training. In data-augmented training, we used 200 cuberoot Mel features as input which yielded better results. For the NMSE loss function, we used a weight of $w_u = T_u$ for each utterance, where T_u is the utterance length in frames. For the SNR loss, we used a weight of $w_u = 1$ for each utterance. For the SNR loss, we use a $\tanh$ compression with $A = 20$. The last row in the table contains the complex IRM (cIRM) prediction result when we use the loss function along with the compression suggested in [16, 17] with data-augmented training. Using the MSA target usually tended to achieve a better PESQ score as compared to the PSA target. Note that we used mask transformations as described in Section 4.4 which shrinks MSA masks at test time.

Table 1: *Evaluation results on the CHiME-2 evaluation data set with left channel audio and with original and augmented training data and different loss functions. 100 or 200 dimensional cuberoot-Mel-filterbank features were used as input.*

Method	Loss	SDR	SIR	PESQ	STOI
No enh.	n/a	2.34	2.34	1.55	0.82
Original training data, 100 dim cuberoot-Mel features					
Best in [15]	MSE-PSA	14.51	19.78	2.78	0.91
BLSTM 2x400	MSE-PSA	14.21	19.56	2.70	0.911
CNN+BLSTM	MSE-PSA	14.52	20.23	2.79	0.916
Augmented training data, 200 dim cuberoot-Mel features					
BLSTM 2x400	MSE-PSA	14.62	19.94	2.81	0.915
BLSTM 2x400	MSE-MSA	14.48	18.62	2.81	0.920
BLSTM 2x400	SNR-PSA	14.88	20.23	2.88	0.920
BLSTM 2x400	SNR-MSA	15.03	20.43	2.93	0.923
CNN+BLSTM	MSE-PSA	15.03	21.19	2.94	0.921
CNN+BLSTM	MSE-MSA	14.72	19.71	2.96	0.924
CNN+BLSTM	NMSE-PSA	14.94	21.01	2.89	0.919
CNN+BLSTM	NMSE-MSA	14.85	20.50	2.93	0.920
CNN+BLSTM	SNR-PSA	**15.23**	**21.37**	2.98	0.924
CNN+BLSTM	SNR-MSA	15.11	20.84	**3.00**	**0.925**
CNN+BLSTM	MSE-cIRM	13.89	19.11	2.90	0.919

7. Conclusions

We experimented with on-the-fly data augmentation and novel loss functions for training speech-background separation models. We showed that data augmentation is quite beneficial in training speech-background separation networks. We proposed new loss functions that yield better performance as compared to the typical MSE loss function that was being used in the literature. The new loss functions are shown to outperform old ones while training with on-the-fly data augmentation on CHiME-2 data set as compared to the best result published in the literature. In our future work, we plan to work on larger databases where one can build a generic speech background separator that would work across many different scenarios.

[1]We plan to provide more information about this network architecture in another publication.

8. References

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[2] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 1581–1585.

[3] F. J. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *GlobalSIP Machine Learning Applications in Speech Processing Symposium*, 2014.

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

[5] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 241–245.

[6] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Spoken Language Proceesing, ISCA International Conference on (INTERSPEECH)*.

[7] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[8] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. of ICASSP*, Brisbane, Australia, 2015.

[9] 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. 1849–58, 2014.

[10] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. of ICASSP*. IEEE, 2013, pp. 7092–7096.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[13] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[14] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," *Proc. Interspeech 2017*, pp. 2386–2390, 2017.

[15] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Deep recurrent networks for separation and recognition of single-channel speech in nonstationary background audio," in *New Era for Robust Speech Recognition*. Springer, 2017, pp. 165–186.

[16] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, 2016.

[17] D. S. Williamson and D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1492–1501, 2017.

[18] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 126–130.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] S. J. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 159–164.

[21] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[22] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.