INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Adversarial Network Bottleneck Features for Noise Robust Speaker Verification

Hong Yu[1,2], *Zheng-Hua Tan*[2], *Zhanyu Ma*[1*], *Jun Guo*[1]

[1]Beijing University of Posts and Telecommunications, China
[2]Aalborg University, Denmark

hongyu@bupt.edu.cn, zt@es.aau.dk, mazhanyu@bupt.edu.cn, guojun@bupt.edu.cn

Abstract

In this paper, we propose a noise robust bottleneck feature representation which is generated by an adversarial network (AN). The AN includes two cascade connected networks, an encoding network (EN) and a discriminative network (DN). Mel-frequency cepstral coefficients (MFCCs) of clean and noisy speech are used as input to the EN and the output of the EN is used as the noise robust feature. The EN and DN are trained in turn, namely, when training the DN, noise types are selected as the training labels and when training the EN, all labels are set as the same, i.e., the clean speech label, which aims to make the AN features invariant to noise and thus achieve noise robustness. We evaluate the performance of the proposed feature on a Gaussian Mixture Model-Universal Background Model based speaker verification system, and make comparison to MFCC features of speech enhanced by short-time spectral amplitude minimum mean square error (STSA-MMSE) and deep neural network-based speech enhancement (DNN-SE) methods. Experimental results on the RSR2015 database show that the proposed AN bottleneck feature (AN-BN) dramatically outperforms the STSA-MMSE and DNN-SE based MFCCs for different noise types and signal-to-noise ratios. Furthermore, the AN-BN feature is able to improve the speaker verification performance under the clean condition.

Index Terms: speaker verification, STSA-MMSE, DNN based speech enhancement, adversarial training, bottleneck features

1. Introduction

Recently, generative adversarial networks (GANs) [1] have attracted a tremendous amount of attention and they are successfully applied to many signal generation tasks, such as image generation [2] and image to image translation [3] [4] [5]. A GAN is composed of two networks: a generative network (GN) and a discriminative network (DN). The GN is trained to generate 'fake' data from random inputs and make the generated 'fake' data similar to the 'real' data. The DN is trained to distinguish between the 'fake' and 'real' data. By training these two networks in turn, the generated 'fake' data become more and more similar to the 'real' data. The GAN methodology is an instance of the broader machine learning concept called adversarial training, in which several networks learn together toward competing objectives, resulting in adversarial networks (ANs). An example application of ANs is dialogue generation [6].

So far in the area of audio and speech processing, ANs have received comparatively less attention than they have in image processing. However, some notable exceptions have been published recently. For example a phone/senone classifier is trained by adversarial learning methods in [7][8], and an AN is used for music generation in [9].

In this work, we study ANs to address a well-known problem in speech processing, namely the significant degradation of performance of speech systems under noisy environments. In order to improve the robustness of these systems, in the literatures, a variety of speech enhancement methods are used to recover the clean speech signal from a noisy one, such as a priori Signal-to-noise ratio (SNR) estimation based Wiener filter [10], short-time spectral amplitude minimum mean square error (STSA-MMSE) [11] and non-negative matrix factorization (NMF) [12]. Many deep neural network (DNN) based methods have also been exploited. In [13][14][15], DNNs are used to enhance speech directly by obtaining a denoised time-frequency representation. In [16][17], an ideal time-frequency binary mask (IBM) or ideal time-frequency ratio mask (IRM) is estimated by DNNs firstly and is then used to recover clean speech.

In this paper, we propose a non-task-specific adversarial network for extracting bottleneck features (AN-BN). Similar to GANs, the AN-BN extractor also includes two cascade connected networks, an encoding network (EN) and a discriminative network (DN). Unlike GAN using random inputs, the AN uses clean and noisy acoustic features as training data and noise types as training labels. The EN is trained to produce AN-BN features which are invariant to noise types and the DN is trained to distinguish the types of additive noises. By training them in turn, noise robust AN-BN features are produced by the EN.

The proposed AN-BN features are applied to speaker verification (SV). As we know, the performance of classical SV systems, such as Gaussian Mixture Model-Universal Background Model (GMM-UBM) [18] and i-Vector systems [19], greatly degrades when speech signals are corrupted by additive noises [20]. Many works have been done on developing noise robust SV systems during last decades [21]. In the back end, pooling clean and noisy speech together to train SV systems is able to make the trained model better fit the noisy conditions [22][23]. In the front end, a variety of speech enhancement methods, e.g., Wiener filter [10], STSA-MMSE [11]and DNN speech enhancement [11][13][15][16] are used. For the comparison purpose, the STSA-MMSE and DNN speech enhancement (DNN-SE) front ends are chosen as baseline front ends for a text-dependent SV system under different noise conditions.

The paper is organized as follows. In Section 2, we introduce the structure of the proposed AN-BN feature extractor and the training method. In Section 3, we introduce two baseline frontends, STSA-MMSE and DNN-SE for the comparison purpose. In Section 4, the speech corpora and noise data used for AN training and SV evaluation are described. In Section 5, the experimental design and results are presented, and finally the paper is concluded in Section 6.

* The corresponding author is Zhanyu Ma

http://dx.doi.org/10.21437/Interspeech.2017-883

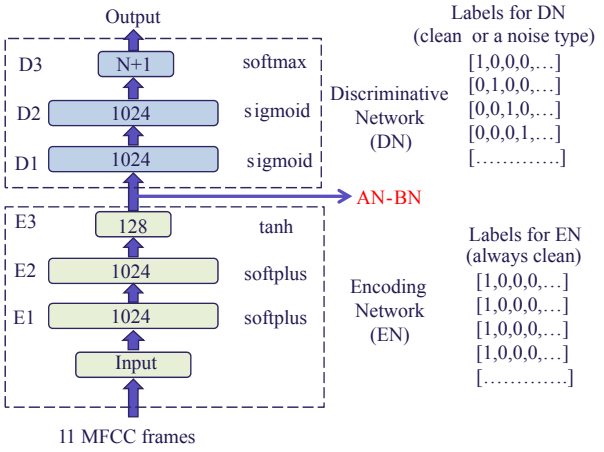

Figure 1: *The structure of AN bottleneck feature extractor.*

2. AN-BN feature extractor

The proposed AN-BN feature extractor consists of two cascade connected networks, an EN and a DN, as shown in Fig. 1. The EN includes three hidden layers, E1, E2 and E3, with 1024, 1024 and 128 nodes, respectively. Following the suggestion in [24], the activation functions of E1 and E2 are both chosen as softplus ($\log(\exp(x)+1)$) and tanh is selected as the activation function of E3.

The input to the EN is batch normalized mel-frequency cepstral coefficients (MFCCs) of 11 frames including as context five past frames and five future frames, and the output of E3 is used as the AN-BN feature. The DN includes two sigmoid hidden layers with 1024 nodes each and a softmax output layer. The dimension of the output layer is $N+1$, representing N noise types and clean.

When training the DN, noise types are used as training labels, and we update the parameters θ_D of the DN only, while keeping the values of parameters θ_E of the EN unchanged. When training the EN, the label 'clean speech' is used for all inputs so as to output noise-invariant features, and we update θ_E only, while keeping the values of θ_D unchanged.

Clean and noisy training data are randomly grouped into small batches with 32 utterances each, and stochastic gradient descent (SGD) is used to train the EN and DN. The number of training epochs is selected as 30.

The cross entropy function is selected as the cost function as shown in equations (1) and (2), where x_i means the input feature, m means the number of frames in each mini-batch, L_{E_i} and L_{D_i} stand for the training labels of the i-th frame, used for EN and DN training, respectively.

$$\min_{\theta_E} loss_E = -\frac{1}{m}\sum_{i=1}^{m} L_{E_i} \log[DN(EN(x_i))]^T, \quad (1)$$

$$\min_{\theta_D} loss_D = -\frac{1}{m}\sum_{i=1}^{m} L_{D_i} \log[DN(EN(x_i))]^T. \quad (2)$$

3. Baseline systems

In this section, we introduce two baseline front-ends, STSA-MMSE and DNN-SE. We also describe the GMM-UBM based SV baseline system which will be used to evaluate the performances of different front ends. The GMM-UBM method is chosen as it performs well for short utterances [25][26], which is the case in this paper.

3.1. STSA-MMSE

STSA-MMSE is a noise independent speech enhancement method which does not need the apriori knowledge of noise type or noise level. It is a statistical method which relies on the assumption that discrete Fourier transform (DFT) coefficients of noise free speech follow a generalized gamma distribution [11]. In the STSA-MMSE method, the priori SNR is estimated by the Decision-Directed approach [27] and the noise power spectral density (PSD) is estimated by the noise PSD tracker reported in [28]. For each utterance, the noise tracker is initialized using a noise PSD estimate based on the first 1000 samples.

3.2. DNN based speech enhancement

The IRM estimation based DNN-SE method introduced in [16] is used as another baseline front-end. Following the suggestion in [16], the time-frequency (T-F) representation used to construct the IRM is based on a gammatone filter bank with 64 filters linearly spaced on a Mel frequency scale and with a bandwidth equal to one equivalent rectangular bandwidth (ERB) [29]. The output of each filter bank channel is divided into 20 ms frames with 10 ms overlap. IRM of noisy speech is used as the training label. On the n-th frame of channel ω, IRM can be computed as follows,

$$\mathbf{IRM}(n,\omega) = \left(\frac{\|x(n,\omega)\|^2}{\|x(n,\omega)\|^2 + \|d(n,\omega)\|^2}\right)^{0.5}, \quad (3)$$

where $\|x(n,\omega)\|^2$ means the energy of clean speech of channel ω on the n-th frame and $\|d(n,\omega)\|^2$ stands for the energy of noisy speech of channel ω on the n-th frame. So the label dimension of each training feature frame is 64.

The input to the DNN is a combination of features including 31 MFCCs, 15 amplitude modulation spectrogram (AMS), 13 relative spectral transform perceptual linear prediction (RASTA-PLP) and 64 Gammatone filter bank energies (GFE). Delta and double delta features are computed and a context of 2 past and 2 future frames is utilized, so the dimension of training features is $(31+15+13+64)\times 3\times 5 = 1845$. All feature vectors are normalized to zero mean and unit variance.

The DNN for IRM estimation includes three hidden layers of 1024 nodes. The activation functions for the hidden layer are rectified linear units (ReLUs) [30] and a sigmoid function is for the output layer. The values of the parameters are updated using the SGD approach and the mean square error (MSE) is chosen as the cost function. The number of training epochs is selected as 30.

The trained DNN is used to estimate IRM for test speech, and the estimated IRM is used to reconstruct the T-F representation of enhanced speech. All T-F units in each frequency channel are then concatenated and all overlapping parts are summed. A time domain enhanced speech signal can be synthesized by compensating for the different group delays in the different frequency channels and adding 64 frequency channels [29].

3.3. Speaker verification systems

In this paper we use the classical GMM-UBM SV method to evaluate the performance of three different front-ends. The GMM-UBM based SV system is built and tested in three steps. First, a universal background GMM model (UBM) is trained by

an expectation-maximization algorithm using a large amount of general speech data. Secondly, enrollment speaker GMMs are created using maximum-a-posteriori (MAP) adaptation of the UBM. Finally, the SV score of test speech is computed as the log-likelihood ratio between the claimed speaker's GMM and the UBM. Usually, only clean or enhanced clean enrollment speech data are used for speaker model training. Motivated by the *multi-condition* training method introduced in [22][23][31], we also investigate the performance of *multi-condition* speaker models which are trained by enhanced clean and noisy speech.

4. Speech corpora and noise data

4.1. Speech corpora

4380 male speaker utterances from the TIMIT corpus [32] are used for UBM training. The clean speech data used for training AN, DNN-SE and speaker models and for testing SV are all from RSR2015 corpus [33] as detailed in Table 1. A text-dependent SV system is constructed for 49 male speakers. For training speaker models, text ID 1 and sessions 1, 4, and 7 from male speakers from $m002$ to $m050$ are selected, and for SV testing, sessions 2, 3, 5, 6, 8 and 9 are used. There are in total $49 \times 6 = 294$ utterances used for testing and the trial protocol consists of $49 \times 294 = 14406$ trials.

Table 1: *Male-speaker speech used for training AN, DNN-SE and speaker models and for testing SV.*

System	Text ID.	Sess. ID	Sprk. ID
AN-BN train	2-30	1,4,7	51-100
DNN-SE train	2-30	1,4,7	51-100
Spkr. Model train	1	1,4,7	2-50
SV test	1	2,3,5,6,8,9	2-50

The AN and DNN-SE model are trained using text IDs $2-30$ and sessions 1,4 and 7 from male speakers from $m051-m100$.

Speech used for AN, DNN-SE model and speaker model training was recoded by Samsung Nexus smart phone. SV testing speech was recoded by Samsung Galaxy S and a HTC Desire smart phone, which can make an unmatched microphone/recording setting.

4.2. Noises and noisy speech

In order to simulate the real-life speaker verification scenarios, we consider five different types of noises: Babble, Cantine, Market, Airplane and white Gaussian noise (White). White was generated in MATLAB, Babble was made by adding 6 random speech samples from the Librispeech database [34], Cantine noises were recoded by the authors. Market and Airplane noises were collected by Fondazione Ugo Bordoni (FUB) and are available on request from the OCTAVE project [35]. All noise data are split into three non-overlapping parts for noisy speech generation, which are used in AN and DNN-SE model training, *multi-condition* speaker model training and SV testing, respectively.

Noisy speech is created by taking out a random segment of noise which matches the length of the speech signal, scaling the amplitude of the noise segment to desired SNR levels, and adding it to the speech. The scaling factor is calculated using the ITU speech voltmeter [36].

5. Experimental results and discussion

In order to evaluate the performance of the AN-BN feature for SV, six versions of AN-BN features are investigated: five noise specific AN-BN (NS-AN) features, one for each noise type, and one noise general AN-BN (NG-AN) feature. NS-ANs are trained by clean speech and one particular noisy speech and NG-AN is trained by a combination of clean and all five types of noisy speech.

MFCCs used for the AN training are generated using a 20ms frame length and 10ms frame shift. Energy based voice activity detection (VAD) method is used to delete non-speech frames. The dimension of MFCCs is 57 (without the 0-th coefficient, including static, delta and double delta features), so the input layer of the AN-BN extractor has $57 \times 11 = 627$ nodes.

Because the DN converges faster than the EN, in order to balance the training of EN and DN, the AN training uses noisy speech with high SNRs 10dB and 20dB, which can not be easily distinguished from clean speech. Furthermore, in each mini-batch training, we update the EN three times and update the DN with a 50% probability only.

The same as the AN-BN front end, we also investigate five noise specific DNN-SE (NS-DNN) front ends and one noise general DNN-SE (NG-DNN) front end which are trained by one particular noisy speech and a combination of five types of noisy speech, respectively. Clean and corresponding noisy speech are used for computing labels for training. SNRs of noisy speech used for training DNN-SE models are also 10dB and 20dB.

For evaluating the basic front end (no enhancement) and STSA-MMSE and DNN-SE front ends, MFCCs of 57 dimensions (the same as for AN training) are used for training and testing the SV systems. For the AN-BN front end, the SV system is trained and tested using AN-BN features with 128 dimensions. The mixture number of GMMs is chosen as 512.

SV systems built on different front ends are evaluated in different noise conditions with SNRs ranging from 0dB - 20dB. The system is also tested on the enhanced clean speech in order to investigate the effect of noise robust front ends on the noise free condition.

Firstly, we investigate the performance using *clean* speaker models. For no enhancement front end, clean speech is used for training speaker models, and for other front ends, enhanced clean speech is used, which means each *clean* speaker model is trained by three utterances. Equal error rates (EER) are used to evaluate the performances of different font ends and the results are shown in Table 2.

It can be seen that AN-BN and DNN-SE front ends outperform the STSA-MMSE front end. NS-AN and NG-AN front ends achieve the lowest EERs for the majority of the test conditions. Comparing with the DNN-SE front ends, AN-BN front end can decrease average EERs by about 25% on White and Babble noise and about 40% on the other three noise types. Especially, on SNRs from 0dB to 5dB which are not used for training DNN-SE and AN models, NS-AN and NG-AN perform much better than NS-DNN and NG-DNN, respectively.

Thereafter, we investigate the SV performances under the *multi-condition* training framework. For noise specific situations, enhanced clean speech and one type of enhanced noisy speech with SNR 10dB and 20dB are used for training speaker models, which means each speaker model is trained by nine utterances. For noise general situation, enhanced clean speech and all five types of enhanced noisy speech with SNR 10dB and 20dB are used, each *multi-condition* speaker model is trained by 33 utterances. About no enhancement and STSA-MMSE front

Table 2: *EER (%) of the SV system using different methods for different noise types and SNRs (dB) on clean speaker model.*

noise	SNR	No Enh.	MMSE	NS-DNN	NG-DNN	NS-AN	NG-AN
White	00	45.90	30.95	39.46	40.14	**25.69**	27.02
	05	43.20	21.17	20.75	21.77	**17.01**	17.81
	10	34.61	13.95	**9.86**	10.88	10.24	11.35
	15	26.28	10.20	7.82	8.16	**6.48**	7.51
	20	16.91	8.50	6.12	6.80	**4.42**	5.29
	clean	6.99	5.80	6.02	5.67	3.84	**3.41**
	mean	28.98	15.10	15.01	15.57	**11.28**	12.07
Babble	00	19.05	29.04	17.01	**16.67**	19.03	17.87
	05	14.63	20.40	10.54	10.39	10.20	**9.86**
	10	11.69	12.59	7.82	7.50	**5.44**	**5.44**
	15	11.04	7.82	6.46	6.34	3.21	**3.06**
	20	9.18	6.29	6.12	5.78	**3.06**	3.40
	clean	6.99	5.80	5.78	5.67	**3.00**	3.41
	mean	12.10	13.66	8.96	8.73	7.32	**7.17**
Cantine	00	20.72	19.09	18.71	19.94	**9.18**	9.81
	05	19.20	12.37	8.58	9.18	**5.10**	5.86
	10	14.74	8.16	6.12	6.12	**3.60**	4.44
	15	11.81	6.80	5.49	5.78	3.40	**3.06**
	20	8.50	6.12	5.31	5.44	**3.25**	3.40
	clean	6.99	5.80	5.10	5.67	4.08	**3.41**
	mean	13.66	9.72	8.22	8.69	**4.77**	5.00
Market	00	29.40	25.51	21.43	21.77	**14.29**	14.43
	05	20.07	17.35	9.86	10.59	**6.80**	7.82
	10	15.00	11.90	6.88	7.48	**4.08**	4.42
	15	11.96	8.28	6.46	6.22	**3.06**	3.64
	20	8.93	7.35	5.78	5.76	**3.13**	3.40
	clean	6.99	5.80	5.92	5.67	3.74	**3.41**
	mean	15.39	12.70	9.39	9.58	**5.85**	6.19
Airplane	00	21.09	17.69	16.99	15.99	**9.86**	**9.86**
	05	15.99	12.58	10.55	8.99	**6.12**	6.46
	10	13.61	8.17	7.48	6.12	**4.35**	5.10
	15	11.66	6.53	6.99	6.12	**3.74**	4.08
	20	9.18	6.27	6.15	5.58	**3.06**	3.63
	clean	6.99	5.80	6.12	5.67	**3.40**	3.41
	mean	13.08	9.51	9.05	8.08	**5.09**	5.42

Table 3: *EER (%) of the SV system using different methods for different noise types and SNRs (dB) on multi-condition speaker model.*

noise	SNR	No Enh.	MMSE	NS-DNN	NG-DNN	NS-AN	NG-AN
White	00	35.88	30.95	27.21	26.19	**22.04**	26.87
	05	24.40	20.07	**9.52**	11.22	13.95	17.69
	10	18.37	7.48	**6.12**	7.14	8.42	11.19
	15	15.81	6.46	**5.02**	5.10	5.80	7.50
	20	14.97	6.46	4.65	**4.08**	4.42	5.10
	clean	5.85	4.76	5.78	4.00	2.04	**1.33**
	mean	19.21	12.70	9.72	9.62	**9.44**	11.61
Babble	00	21.77	33.50	16.26	**16.00**	17.35	18.71
	05	15.37	23.13	9.52	**9.18**	10.48	9.86
	10	11.93	16.23	6.99	5.44	5.27	**5.13**
	15	9.52	12.63	6.08	4.76	**3.06**	3.39
	20	8.16	8.84	5.78	4.08	2.72	**2.61**
	clean	6.12	7.12	5.17	4.00	2.38	**1.33**
	mean	12.15	16.91	8.30	7.19	6.88	**6.84**
Cantine	00	24.11	19.05	12.93	11.61	**8.77**	10.20
	05	17.22	12.59	5.91	5.78	5.78	**5.44**
	10	12.93	8.21	4.42	5.10	4.10	**3.75**
	15	10.88	6.91	4.25	4.57	3.74	**3.40**
	20	9.18	6.12	4.27	4.08	3.59	**2.38**
	clean	7.48	6.32	3.78	4.00	2.50	**1.33**
	mean	13.63	9.87	5.93	5.86	4.75	**4.42**
Market	00	36.05	29.25	19.33	18.37	**15.31**	**15.31**
	05	26.06	21.07	**8.16**	**8.16**	8.50	8.50
	10	18.37	13.95	6.24	5.78	**5.29**	**5.29**
	15	13.32	10.98	5.41	4.44	3.88	**3.64**
	20	9.18	7.82	4.53	4.42	3.06	**2.72**
	clean	5.44	6.76	4.29	4.00	2.29	**1.33**
	mean	18.07	14.97	7.99	7.53	6.39	**6.13**
Airplane	00	32.28	25.51	14.78	11.38	11.56	**10.54**
	05	26.87	15.48	8.26	**6.12**	7.69	6.46
	10	21.10	8.16	5.44	**4.78**	6.11	4.95
	15	16.38	6.12	5.53	4.72	4.42	**4.08**
	20	9.86	5.44	4.76	4.23	3.40	**3.00**
	clean	5.83	5.44	4.76	4.00	2.32	**1.33**
	mean	18.72	11.03	7.26	5.87	5.92	**5.06**

end, only noise specific situations are considered.

From the experimental results shown in Table 3, it can be observed that *multi-condition* trained speaker models can improve the performance of SV systems. AN-BN front ends still get the best results for most of the test conditions.

It is surprising to observe that the NG-AN front end outperforms NS-AN for several SNRs and noise types, which means in *multi-condition* SV systems, more speaker model training utterances can help the learned model to fit complex noisy environments and improve the robustness of SV systems.

It is also found that under high SNRs and clean conditions, the AN-BN front end performs much better than the DNN-SE front end. A reasonable explanation to this is that, during the AN training, in the EN updating step, *clean* speech data from different sessions are all trained using the same 'clean' label. It means the AN can extract not only the common information between clean and noisy speech, but also that of of different *clean* speech data. The DNN-SE method, however, sets the training target as recovering the clean speech from the corresponding noisy speech, but it does not train on *clean* speech. That is why EERs of the DNN-SE front end are very similar to the no enhancement front end on clean condition and the AN-BN front end is able to greatly improve the SV accuracy. Generally, comparing with the DNN-SE front end, the AN-BN front end performs better for the SV task. The dimension of the AN-BN front end is 128 while that of the DNN-SE is 57, so the models for the AN-BN front end have a higher complexity. Future work includes reducing the dimension of the AN-BN front end to 57 for a fair comparison using principal component analysis or making the final output of the EN 57 dimensions.

6. Conclusions

In this paper, we proposed a new adversarial networks (AN) based noise robust feature extractor, which consists of two cascade connected networks, one encoding network (EN) and one discriminative network (DN). The EN and DN are trained in turn and the outputs of the EN are used as robust features for speaker verification (SV). When training the DN, the values of EN parameters are kept unchanged and noise types are used as training labels. When the EN is trained, the values of DN parameters are kept unchanged and all input speech data are assigned the same label, namely the clean speech label. Being trained using clean and noisy speech, the AN bottleneck (AN-BN) features can not only gain the common information between noisy and clean speech, the common information among clean speech recoded in different sessions can also be extracted. This trait makes the AN-BN features particularly suitable for the noise roust SV task. Experimental results on the RSR2015 data base show that the AN-BN front end outperforms short-time spectral amplitude minimum mean square error (STSA-MMSE) and deep neural network base speech enhancement (DNN-SE) front ends for the majority of the tested conditions, especially on high signal to noise ratios (SNR) and clean conditions. In the future, we will conduct more extensive comparison with existing methods and evaluate the performance of the AN-BN features on other speech applications under noisy conditions, e.g., speech recognition and spoofing detection.

7. Acknowledgements

This work is partly supported by Project 61402047, 61273217, 61401259 supported by NSFC, Beijing National Science Foundation No. 4162044, Scientific Research Foundation for Returned Scholars, Ministry of Education of China, Chinese 111 program of Advanced Intelligence, Network Service Grant B08004. OCTAVE Project (No. 647850), funded by the Research European Agency (REA) of the European Commission, in its framework programme Horizon 2020. The authors would like to thank Morten Kolbæk for his assistance and software used for the DNN speech enhancement baseline systems.

8. References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] G.-J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *arXiv preprint arXiv:1701.06264*, 2017.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[5] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[6] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547*, 2017.

[7] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, "Invariant representations for noisy speech recognition," *arXiv preprint arXiv:1612.01928*, 2016.

[8] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," *Interspeech 2016*, pp. 2369–2372, 2016.

[9] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[10] P. Scalart *et al.*, "Speech enhancement based on a priori signal to noise estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 1996, pp. 629–632.

[11] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1741–1752, 2007.

[12] N. B. Thomsen, D. A. L. Thomsen, Z.-H. Tan, B. Lindberg, and S. H. Jensen, "Speaker-dependent dictionary-based speech enhancement for text-dependent speaker verification," *Interspeech 2016*, 2016.

[13] O. Plchot, L. Burget, H. Aronowitz, and P. Matëjka, "Audio enhancing with dnn autoencoder for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5090–5094.

[14] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[15] M. Kolbœk, Z.-H. Tan, and J. Jensen, "Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification," in *Spoken Language Technology Workshop (SLT)*, 2016, pp. 305–311.

[16] M. Kolbæk, Z.-H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 25, no. 1, pp. 153–167, 2017.

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

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[20] F. Yan, A. Men, B. Yang, and Z. Jiang, "An improved ranking-based feature enhancement approach for robust speaker recognition," *IEEE Access*, vol. 4, pp. 5258–5267, 2016.

[21] O. Plchot, S. Matsoukas, P. Matějka, N. Dehak, J. Ma, S. Cumani, O. Glembek, H. Hermansky, S. Mallidi, N. Mesgarani *et al.*, "Developing a speaker identification system for the darpa rats project," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6768–6772.

[22] Y. Lei, L. Burget, and N. Scheffer, "A noise robust i-vector extractor using vector taylor series for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6788–6791.

[23] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, "Towards noise-robust speaker recognition using probabilistic linear discriminant analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4253–4256.

[24] M. A. Soumith Chintala, Emily Denton and M. Mathieu, "How to train a GAN? Tips and tricks to make GANs work," http://timmurphy.org/2009/07/22/line-spacing-in-latex-documents/.

[25] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan, "Further Optimisations of Constant Q Cepstral Processing for Integrated Utterance and Text-dependent Speaker Verification," in *Processing of Spoken Language Technology Workshop (SLT)*, 2016.

[26] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[27] 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.

[28] R. C. Hendriks, R. Heusdens, and J. Jensen, "Mmse based noise psd tracking with low complexity," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 4266–4269.

[29] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE Press, 2006.

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[31] H. Yu, A. Sarkar, D. A. L. Thomsen, Z.-H. Tan, Z. Ma, and J. Guo, "Effect of multi-condition training and speech enhancement methods on spoofing detection," in *International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, 2016, pp. 1–5.

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

[33] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[35] M. Falcone, M. C. Benoit Fauve *et al.*, "Corpora collection," *OCTAVE (Objective Control of TAlker VErification), Deliverable 17*, 2016.

[36] G. Recommendation, "191: Software tools for speech and audio coding standardization," *Int. Telecommun. Union, Geneva, Switzerland*, 2005.