Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

DENOISING X-VECTORS FOR ROBUST SPEAKER RECOGNITION

Mohammad MohammadAmini, Driss Matrouf, Paul-Gauthier Noé

LIA (Laboratoire Informatique d'Avignon)

Avignon University

{mohammad.mohammadamini , driss.matrouf , paul-gauthier.noe}@univ-avignon.fr

Abstract

Using deep learning methods has led to significant improvement in speaker recognition systems. Introducing x-vectors as a speaker modeling method has made these systems more robust. Since, in challenging environments with noise and reverberation, the performance of x-vectors systems degrades significantly, the demand for denoising techniques remains as before. In this paper, for the first time, we try to denoise the x-vectors speaker embedding. Our focus is on additive noise. Firstly, we use the i-MAP method which considers that both noise and clean x-vectors have a Gaussian distribution. Then, leveraging denoising autoencoders (DAE) we try to reconstruct the clean x-vector from the corrupted version. After that, we propose two hybrid systems composed of statistical i-MAP and DAE. Finally, we propose a novel DAE architecture, named Deep Stacked DAE, composed of several DAEs where each DAE receives as input the output of its predecessor DAE concatenated with the difference between noisy x-vectors and its predecessor's output. The experiments on Fabiol corpus show that the results given by the hybrid DAE i-MAP method in several cases outperforms the conventional DAE and i-MAP methods. Also, the results for Deep Stacked DAE in most cases is better than the other proposed methods. For utterances longer than 12 seconds we achieved a 51% improvement in terms of EER with Deep Stacked DAE, and for utterances shorter than 2 seconds, Deep Stacked DAE gives 18% improvements compared to the baseline system.

Key terms: Speaker recognition, x-vector, i-MAP, Noise compensation, Denoising autoencoder

1. Introduction

Speaker recognition is the task of identifying speakers from their utterances. In the past decade, introducing the i-vector statistical model and x-vector speaker embedding has led to notable progress in the speaker recognition area. However, x-vector speaker modeling method has caused substantial improvement in speaker recognition system, the performance of this system in challenging environments with the presence of unseen noises and reverberation degrades significantly. In our experiments, with low SNR and a large number of unseen noises added to the test data, we observed that the performance of x-vector embedding diminishes drastically in comparison to results obtained with noise-free x-vectors.

Some studies [1, 2, 3] showed that by increasing the number of speakers, the amount of training data, and by using data augmentation, the x-vectors can achieve a certain degree of robustness with the presence of noise, but this degree of robustness remains very insufficient especially in the case of low SNR.

In this paper, in addition to the common data augmentation techniques, we propose to denoise the noisy x-vectors to approach the performance obtained by noise-free x-vectors. In this manner, it becomes easier to target a specific unseen noise or to adapt the denoising system to new conditions. In other words, the proposed system is a pipeline of two systems, the first one allows to generate the best x-vectors possible and the second is used to denoise it.

Applying denoising techniques at the speaker modeling level has been done successfully in the i-vector space [4, 5, 6]. In this paper we apply statistical denoising techniques on x-vectors that works effectively in i-vector domain. Although, we want to explore the effectiveness of DNN denoising techniques in x-vector domain. Our first attempt consists of using a statistical approach based on the use of maximum a posteriori (MAP), namely the i-MAP approach [5]. The i-MAP denoising technique has been used successfully in the i-vector space. Then, we compare the results obtained by i-MAP with denoising autoencoder. Furthermore, we propose two hybrid systems that use both denoising autoencoders and i-MAP. Finally, we propose a novel DNN, named Deep Stacked DAE that outperforms all the other methods.

Our contributions are:

- We applied i-MAP statistical technique in the x-vector space.
- We introduced two hybrid DAE architectures combined with i-MAP to denoise x-vectors
- We also introduced another DAE architecture, composed of several DAEs, that each DAE receives the output of its predecessor DAE concatenated with the difference between noisy x-vectors and its predecessor's output.

In all denoising techniques used in this paper, our goal is to develop a system that tries to compensate for different kinds of noises without explicit information about the noise for a given utterance in the test data. We expect that by including explicit information about the noise (obtained from non-speech part of the test segment) it is possible to achieve more robust systems with denoising autoencoders and our next exploration will be focused in that direction.

10.21437/Odyssey.2020-11

The proposed techniques only compensate for additive noise, but the same idea can be applied to the short duration variabilityy or to the reverberation. Of course, in these cases, the architecture of the different models must be adapted for modeling the specific characteristics of such variabilities.

2. Related works

The previous work on denoising techniques for speaker recognition has been done in three levels: signal level, feature level, and speaker modeling level (i-vectors, x-vectors). Thanks to its statistical properties, developing denoising techniques in modeling level is more promising and easier than signal or feature level. The relative improvement of EER, in signal level [7], shows that results obtained in speaker modeling level are better than signal or feature level [4].

In the signal level, statistical and deep learning speech enhancement methods are used. Statistical speech enhancement methods consider some frames from the beginning of utterance as the noise information. Since these methods do not consider the time-varying nature of the noise, they are not so effective [8, 9]. Besides statistical models, several deep learning speech enhancement methods are used in speaker recognition. In [7], two different DNN architectures were proposed in order to denoise the corrupted signal. In the first one, the log-magnitude spectrum passed through a five-layer BLSTM network. The hidden layers are used to handle the context dependencies in the signal and the output LSTM layer is used to reconstruct the clean signal. In another architecture, a CNN encoder-decoder network is used to denoise the short-time Fourier transform (STFT) magnitude spectrogram. Denoising autoencoder is another DNN method used as a preprocessing step in speaker recognition. In [10] a deep feedforward autoencoder trained in order to transform the noisy log magnitude spectrum to a clean compact encoded signal in the same domain. In another paper, a DNN architecture that uses the feedback from the speaker verification (SV) system to generate a ratio mask was designed. The generated mask multiplied pointwise with the original spectrogram to remove unnecessary parts of the signal for speaker verification [11].

Several publications deal with denoising techniques at speaker modeling level. In [12], a new speaker embedding method called adversarial speaker verification (ASV) is proposed. ASV tries to not only classify the utterances from speaker embedding but also to classify the type of utterance in terms of environmental noise, and Signal to Noise Ratio (SNR). I-MAP is a statistical denoising method that is applied in the i-vector space. The main advantage of this method is that it uses both information about the relation between noisy and clean speech and the clean speech distribution [5]. A nonparametric algorithm without considering the relation between corrupted and clean i-vector was proposed by [4], that utilizes the joint distribution of corrupted and clean i-vectors to denoise corrupted i-vector with an MMSE estimator. Neural autoencoders for denoising i-vectors have also been investigated in [13] and in [14] a classifier is jointly trained with a DAE in order to make the new i-vector more discriminative.

To the best of our knowledge, there seems to be no previous attempt to do noise compensation in x-vector framework except ours. In our work we show the performance degradation of the x-vector speaker modeling method in a noisy environment. Then we try to compensate the negative effect of additive noise using different denoising techniques.

This paper is structured as follows: First, our methodology is described in section 3. In section 4, the experiment setup and data are described. Later, section 5 presents the results of the experiments.

3. Methodology

In this section, the details about the i-MAP, DAE, and our proposed variants of DAEs are discussed. Firstly, the i-MAP method is described. Then, denoising autoencoders are described and two hybrid systems developed from i-MAP and DAEs are presented. Finally, a novel DAE architecture is introduced.

3.1 i-MAP

The i-MAP is a statistical denoising method originally developed for the i-vector domain. Since in our work it is used for denoising x-vectors, from now on, we call it x-MAP. We define X and Y as the random variables for clean and noisy x-vectors. A third random variable for noise is defined as:

$$N = Y - X \tag{1}$$

Where:

$$X \sim N(\mu_X, \sum_X) \tag{2}$$

$$N \sim N(\mu_N, \sum_N) \tag{3}$$

In the x-MAP approach, we assume that both noise and clean x-vectors have a Gaussian distribution.

For a given noisy x-vector Y0, from 1, 2, and 3 relations we fined:

$$f(Y0|X) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_N|^{\frac{1}{2}}} exp^{-\frac{1}{2}(Y0-X-\mu_N)^T \sum_N^{-1}(Y0-X-\mu_N)} \tag{4}$$

To find X0, the clean version of Y0, a MAP estimator is used:

$$X0 = argmax_X\{ln\, f(Y0|X)f(X)\} \tag{5}$$

The final expression of calculating denoised x-vectors is:

$$X0 = (\Sigma_N^{-1} + \Sigma_N^{-1})^{-1}(\Sigma_N^{-1}(y - \mu_N) + \Sigma_X^{-1}\mu_X) \tag{6}$$

More details about the mathematics of x-MAP can be found in [4, 5].

3.2 Denoising autoencoder

Denoising autoencoder is a specific type of autoencoder that takes the noisy x-vector as input and tries to reconstruct the clean version at the output. Denoising autoencoder tries to minimize:

$$L(\boldsymbol{x}, f(\boldsymbol{y})) \tag{7}$$

where L is the loss function, $\boldsymbol{y}$ is the corrupted x-vector and $f(\boldsymbol{y})$ is the output of DAE [15].

In this paper, we propose to use denoising autoencoders to denoise x-vectors. Then, we suggest three variations of DAE that are described in the following sections.

3.3 Hybrid systems

Here, we try to benefit from the potential of x-MAP and DAE simultaneously. In order to do this, we combine those systems in two ways:

DAE x-MAP: In this system, we used DAE and x-MAP methods sequentially. Firstly, the noisy x-vector is denoised by leveraging DAE, then the output of the DAE is denoised with x-MAP. The architecture of this system is presented in Figure 1.A.

Gaussian DAE: Adding a regularization term to the loss function is a common way to constraint the solution space in denoising DNNs [16, 17]. In Bayesian formulation, the regularization terms correspond to prior probabilities added to the loss function. Similar to x-MAP, in Gaussian DAE, we want to impose on the output of DAE and the estimated noised to have a Gaussian distribution.

In the same manner, we add a regularization term to the MSE loss function. The proposed system is named Gaussian DAE. In Gaussian DAE we defined a new loss function to reduce MSE between input and output, and simultaneously maximize the a priori probabilities of the noise and of the obtained x-vectors (both assumed to be Gaussian) like in the x-MAP. In this system, the following relation is used as a loss function:

$$Loss = MSE + (Y - X - \mu_N)\Sigma_N^{-1}(Y - X - \mu_N) + \{(Y - X - \mu_X)\Sigma_X^{-1}(Y - X - \mu_X)\} \quad (8)$$

Where, Y is the input of DAE and X is the output of DAE at each mini-batch, μ_N and μ_X are the mean of noise and clean x-vectors respectively, Σ_N^{-1} and Σ_X^{-1} are the inverse of the covariance matrix for noise and clean data, all these parameters are estimated on the whole training data. The architecture of this system is described in Figure 1.B.

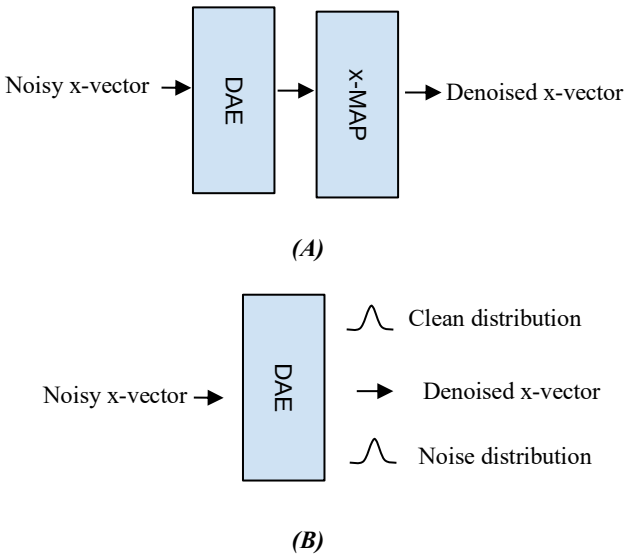

Figure 1. *(A) the hybrid DAE x-MAP; (B) Gaussian DAE: Impose on DAE that assumes denoised x-vectors and the noise have a Gaussian distribution.*

3.4 Deep stacked Denoising Autoencoder

In this subsection, we introduce a new denoising autoencoder called Deep Stacked DAE. In this architecture we have several DAE blocks. The noisy x-vectors fed to the first DAE. The next DAE block receives X_i (the output of its predecessor block) concatenated with $Z_i = Y - X_{i-1}$(the difference between noisy x-vectors and the output of the previous block). The stacked DAEs are trained jointly with the SGD optimization algorithm. The architecture of Deep Stacked DAE is presented in Figure 2.

As we can see in next section, this architecture outperforms all other methods. One assumption behind the effectiveness of this method is that using the difference between noisy x-vectors and the output of the previous DAE, can capture the noise information. Let X be a clean x-vector and Y be a noisy x-vector. If we use Y and X-Y in the input of denoising autoencoder and construct $\hat{X}$ (the denoised x-vectors) in the output, the results will be very close to X. Despite the output layer that uses MSE to shift the output toward clean x-vectors, in hidden layers we don't have such kind of implications. So, we don't know about the nature of the output of each DAE. Moreover, this is similar to residual connection [24] which helps to build deeper models.

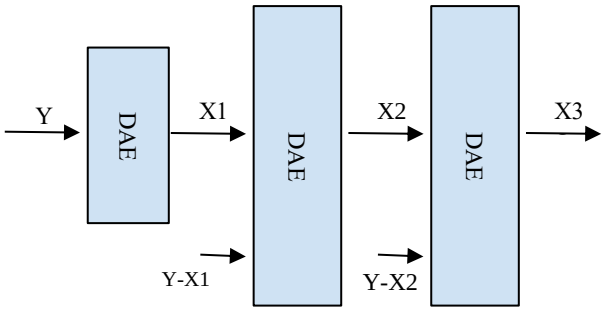

Figure 2*. Deep Stacked DAE*

4. Experiments setup

4.1 Corpus

In this subsection we briefly discuss all datasets used in our experiments:

Voxceleb: The Voxceleb dataset released in two stages: Voxceleb1 and Voxceleb2. Voxceleb1 contains 100,000 utterances from 1,251 celebrities. Voceleb2 includes over 1 million utterances from 6,000 speakers. The speakers are from different ethnicities with different ages, professions and accents. All utterances crucially degraded with different types of noises including background chatter, laughter, overlapping speech and room acoustics. Although there are a lot of variations in recording devices and channels [18].

Musan: The Musan corpus consists 109 hours audio including 60 hours speech, 42 hours music, and 929 noise files. The Musan corpus is suitable for data augmentation [19].

BBC Noise: The BBC Noise corpus contains16,000 sound effects made available [20].

Fabiol: Fabiol is a French corpus consisting of 6882 utterances. The length of files spans from very short utterances less than 2 seconds to very long utterances.

4.2 Train x-vector extractor

In the experiments, a standard Kaldi recipe introduced in [21] is used. Firstly, we augmented Voxceleb1 with different branches of Musan corpus (music, babble, noise, reverberation). Then, we extracted MFCC features for 500,000 randomly chosen utterances from the augmented data. Then, Cepstral Mean Variance Normalization (CMVN) is applied to the features. Finally, the VAD is applied to remove silent frames. In the next step, the TDNN discussed in [21] is used to extract x-vectors. The trained network is used in all experiments to create x-vectors in train and test parts of denoising techniques.

4.3 Train and test x-vectors used in denoising techniques

In denoising techniques, we need the noisy x-vectors and their corresponding clean version. The x-vectors are extracted with the x-vector extractor described in 4.2 section. Simulating noisy signals by adding a noise to clean signals in the time domain is an approach which has proven its effectiveness for generating training data in x-vector extractor (data augmentation). However, in this paper we use the same approach to generate training and test data used in denoising techniques.

For training x-vectors, we used Voxceleb1 and Voxceleb2. Firstly, we extracted the x-vectors from clean files in Voxceleb1 and Voxceleb2. Then, BBC Noises and Musan were added to Voxceleb1 and Voxceleb2 with different SNRs from 0 to 15 and the x-vectors of noisy files extracted. We used 1638 noise files from BBC corpus. The final version of the train data consists 1.975 million pairs of noisy x-vectors and their corresponding clean version. For some clean files there is more than one corrupted version.

For test and enrollment dataset we used the Fabiol Corpus. The Fabiol corpus consists 6882 utterances where half of them were used for enrollment and the remaining part used as the test dataset. Because in real applications it is possible to have clean data for enrollment, in our experiments we used the clean files for enrollment. But for the test part we added 547 different noises form BBC corpus to the test files with different SNRs from 0 to 15. The noise files used in the test are different from those used in the train dataset. Since the length of utterances in Fabiol is varied from very short (less than 2 seconds) to longer utterances (more than 12 seconds), we separated the utterances in 6 groups to investigate the results of denoising methods on each group and specially to observe the effectiveness of the denoising techniques on very short utterances.

4.4 Scoring method:

In the experiments, the PLDA classifier is used. Before training the PLDA, the x-vectors are centered and their dimensionality reduced to 128 with LDA.

5. Results

In this section, we describe the results obtained by statistical x-MAP method and our proposed methods. The results are presented in Table 1 and Table 2. In our experiments, we used the equal error rate (EER) metric to evaluate the performance of the recognition system.

Clean: In this experiment the clean version of x-vectors is used in the test dataset. As it is shown, the results are strongly dependent on the duration of test files. The aim of this experiment is to compare the results obtained by denoising techniques with noise-free x-vectors.

Baseline: To show the weakness of x-vectors in noisy environments, the BBC noise files were added to the test data. In Table 1, we can see that there is a drastic degradation in our results. For example, for utterances longer than 12 seconds the EER increased from 0.833 to 5.131.

x-MAP: In this experiment, we tried to find a denoised version of test x-vectors by using relation 6 in section 2.1. As it is shown in Table 1, in all cases the x-MAP improves the results significantly. For utterances longer than 12 seconds, it gives 50% improvements in terms of EER. It can also be observed that the best gains are obtained for longer segments.

DAE: We did several experiments to find the best denoising autoencoder. In the best architecture we found, three layers are used. The activation function of the input and output layers is linear. In the hidden layer, there are 1024 neurons with Tanh activation function. The network is trained with stochastic gradient descent algorithm with learning rate equal to 0.02 and 0.0001 decay of the learning rate. The number of epochs in this experiment and all other variations of DAEs set to 100, and Mean Square Error (MSE) is used as loss function. From our experiments, we observed that the number of epochs is very important and small reduction in MSE value has a large impact on the results. In Table 1, we can see that for shorter utterances the fine-tuned DAE outperforms x-MAP method but for longer utterances the results are equivalent to x-MAP ones. In all experiments with conventional DAE and its modifications in the next experiments, we used Tensorflow [22] and Keras [23] frameworks.

DAE x-MAP: In this method, we used the DAE with the same architecture and parameters used in simple DAE. The denoised x-vectors are passed through the x-MAP method for further noise compensation. In several cases, the results with this hybrid system are better than simple DAE. In Table 1. we can see that for utterance between 10 and 12 seconds we achieved 46% improvement in relative EER.

Gaussian DAE: In this experiment, we try to not only decrease the MSE with DAE but also maximize the prior probabilities of the estimated noise and of the obtained x-vectors. The only difference between Gaussian DAE and simple DAE is the loss function. In this experiment the function defined in relation 8 were used as loss function. The results in Table 1 show that in some cases, the Gaussian DAE gives better results than x-MAP and conventional DAE.

Deep Stacked DAE: In this experiment, the Deep Stacked DAE which was introduced in section 2.5 is used. In the first DAE, there are three layers. The input and output layers are linear and the hidden layer's activation function is Tanh. The output of the first DAE concatenated with the difference between noisy x-vector and the output layer from its predecessor DAE, fed to the next DAE. In the second DAE there are two Tanh layers with 1024 neurons and the output layer is linear. The number of neurons in the output layer is 512 which equals to the length of noisy x-vector. The network was trained with stochastic gradient descent method. The learning rate is 0.02 and the decay of learning rate is 0.0001.

In another experiment, three stacked DAEs are used. The architecture of the third DAE, is the same as the second one. The third DAE receives its input from the output of the second DAE concatenated with the difference between noisy x-vectors

and the output layer of the second DAE. The results in Table 2 show that adding more DAEs doesn't cause significant improvement.

As we can see from Table 1 and 2, there is a slip in the results for utterances between 8 and 10 seconds. We believe that it happens because the number of trials for this experiment in the Fabiol corpus is small.

Table 1. *Results obtained by x-MAP, DAE, DAE x-MAP, and Gaussian DAE in terms of EER for utterances with different lengths*

Length (second)	**s<2**	**2<s<4**	**4<s<6**	**6<s<8**	**8<s<10**	**10<s<12**	**12<s**
Clean	11.59	7.646	4.144	2.239	3.111	1.538	0.8339
Baseline	15.94	12.88	10.5	7.836	8.889	6.667	5.131
x-MAP	14.2	11.07	8.011	5.597	5.333	4.103	**2.630**
DAE	**13.62**	10.87	8.287	5.597	5.778	4.103	2.694
DAE x-MAP	14.78	10.87	8.287	**4.851**	**5.333**	**3.59**	2.758
Gaussian DAE	14.2	**9.859**	**7.459**	5.597	5.778	4.103	3.143

Table 2. *Results for proposed Deep Stacked DAE*

Length/ N° DAEs	**s<2**	**2<s<4**	**4<s<6**	**6<s<8**	**8<s<10**	**10<s<12**	**12<s**
2	13.04	10.46	8.011	5.224	5.333	3.59	2.502
3	13.04	9.658	7.735	4.851	5.778	4.103	2.502

5. Conclusion and future work

In this paper we applied noise compensation in x-vector space. Firstly, we showed that in x-vector space when there are many unseen noises, the results degrade substantially. Then, we tried to exploit x-MAP statistical denoising technique, originally designed for i-vector space, to denoise x-vectors. The x-MAP technique is applicable in x-vector domain. In this paper, several variations of denoising autoencoders are proposed. The combination of x-MAP and DAE is better than using them separately for denoising x-vectors. In another method (Gaussian DAE), we defined a new loss function which tries to combine the MSE term with two additional terms corresponding to the likelihood of the output data with respect to prior Gaussian distributions. Finally, a Deep Stacked DAE was proposed that each DAE receives the output of its predecessor DAE concatenated with the difference between noisy x-vectors and its predecessor's output. The results obtained by Deep Stacked DAE, outperforms statistical x-MAP technique and other variations of DAE discussed in this paper. We expect that by having an estimation of the noise we can achieve better results; in our future research we try to find explicit information about the noise from non-speech frames in test utterances and use this information alongside noisy x-vectors in denoising autoencoder to design more robust speaker recognition systems.

Acknowledgement

This work was funded by ROBOVOX and VoicePersonae ANR projects.

Bibliography

[1] M, McLaren, D. Castan, M. Kumar Nandwana, L. Ferrer and E. Yilmaz, "How to train your speaker embedding extractor," in *Odyssey 2018*, 2018.

[2] Ondrej Novotny, Oldrich Plchot, Pavel Matejka, Ladislav Mosner, Ondrej Glembek, "On the use of X-vectors for Robust Speaker Recognition," in *Odyssey 2018*, Les Sables d'Olonne, France, 2018.

[3] A. Kanagasundaram, S. Sridharan, G. Sriram, S. Prachi3, C. Fookes, "A Study of X-vector Based Speaker Recognition on Short Utterances," in *Interspeech*, 2019.

[4] Waad Ben Kheder, Matrouf Driss, Moez Ajili, Jean-François Bonastre, "A Unified Joint Model to Deal With Nuisance Variabilities in the i-Vector Space," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[5] Waad Ben Kheder, Driss Matrouf. Pierre-Michel Bousquet. Jean-François Bonastre, Moez Ajili, "Fast

i-vector denoising using MAP estimation and a noise distributions database for robust speaker recognition," *Computer Speech & Language,* vol. 45, pp. 104-122, 2017.

[6] Waad Ben Kheder, Matrouf Driss, Moez Ajili, Jean-François Bonastre, "Probabilistic Approach Using Joint Clean and Noisy i-Vectors Modeling for Speaker Recognition," in *Interspeech 2016*, 2016.

[7] Sefik Emre Eskimeza, Peter Soufleris, Zhiya Duana, Wendi HeinIman, "Front-end speech enhancement for commercial speaker verification systems," *Speech Communication,* vol. 99, pp. 101-113, 2018.

[8] S. Boll, "Suppression of Acoustic Noise in Speech Using Spectral Subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 27, no. 2, pp. 113-120, 1979.

[9] Y. E. a. D. Malah, "Speech enhancement using a minimum mean-square error-log-spectral amplitude estimator(Article)," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 33, no. 2, pp. 443-445, 1985.

[10] Ondřej Novotný, Oldřich Plchot, Ondřej Glembek, Jan Honza Černocký, Lukáš Burget, "Analysis of DNN Speech Signal Enhancement for Robust Speaker Recognition," *Computer Speech & Language,* vol. 58, pp. 403-421, 2019.

[11] Suwon Shon, Hao Tang, James Glass, "VoiceID Loss: Speech Enhancement for Speaker Verification," in *INTERSPEECH 2019*, Graz, Austria, 2019.

[12] Zhong Meng, Yong Zhao, Jinyu Li, Yifan Gong, "Adverserial Speaker Verification," in *ICASSP 2019*, Brighton, 2019.

[13] Timur Pekhovsky, Sergey Novoselov, Aleksei Sholohov, Oleg Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Odyssey 2016*, Bilbao, Spain, 2016.

[14] Shivangi Mahto, Hitoshi Yamamoto, Takafumi Koshinaka, "I-vector Transformation Using a Novel Discriminative Denoising Autoencoderfor Noise-robust Speaker Recognitio," in *INTERSPEECH 2017*, Stockholm, Sweden, 2017.

[15] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.

[16] J. D. Y.-n. W. Li Chai, "Gaussian denisty guided deep neural network for single-channel speech enhancement," in *2017 IEEE International workshop in machine learning for signal processing*, Tokyo, Japan, 2017.

[17] W. Z. S. G. L. Z. Kai Zhang, "Learning Deep CNN Denoiser Prior for Image Restoration," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Arsha Nagrani, Joon Son Chung, Weidi Xie, Weidi Xie, Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language,* vol. 60, 2020.

[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, "MUSAN: A Music, Speech, and Noise Corpus," 28 10 2015. [Online]. Available: https://arxiv.org/abs/1510.08484. [Accessed 25 01 2020].

[20] BBC, "BBC," BBC, [Online]. Available: http://bbcsfx.acropolis.org.uk/. [Accessed 25 01 2020].

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition,," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, 2018.

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, "TensorFlow: Large-scale machine learning on heterogeneous systems," Tensorflow, 2015.

[23] F. Chollet, "Keras," [Online]. Available: https://keras.io/ [Accessed 25 01 2020]..

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," [Online]. Available: https://arxiv.org/abs/1512.03385 [Accessed 07 04 2020]..