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

Generalization Of Audio Deepfake Detection

Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth,
Ganesh Sivaraman, Elie Khoury

Pindrop, Atlanta, GA, USA
{tchen,akumar,pnagarsheth,gsivaraman,ekhoury}@pindrop.com

Abstract

Audio Deepfakes, technically known as logical-access voice spoofing techniques, have become an increased threat on voice interfaces due to the recent breakthroughs in speech synthesis and voice conversion technologies.

Effectively detecting these attacks is critical to many speech applications including automatic speaker verification systems. As new types of speech synthesis and voice conversion techniques are emerging rapidly, the generalization ability of spoofing countermeasures is becoming an increasingly critical challenge. This paper focuses on overcoming this issue by using large margin cosine loss function (LMCL) and online frequency masking augmentation to force the neural network to learn more robust feature embeddings. We evaluate the performance of the proposed system on the ASVspoof 2019 logical access (LA) dataset. Additionally, we evaluate it on a noisy version of the ASVspoof 2019 dataset using publicly available noises to simulate more realistic scenarios. Finally, we evaluate the proposed system on a copy of the dataset that is logically replayed through the telephony channel to simulate spoofing attacks in the call center scenario.

Our baseline system is based on residual neural network, and has achieved the lowest equal error rate (EER) of 4.04% among all single-system submissions during the ASVspoof 2019 challenge. Furthermore, the additional improvements proposed in this paper reduce the EER to 1.26%.

1. Introduction

The fast growing voice-based interfaces between humans and computers have led to the need for more accurate voice biometrics strategies. The accuracy of speaker verification technology has improved by leaps and bounds in the past decade with the help of deep learning. At the same time, the ability to spoof and impersonate voices using deep learning based speech synthesis systems have also significantly improved.

Such high quality text-to-speech synthesis (TTS) and voice conversion (VC) approaches can successfully deceive both humans and automatic speaker verification systems. This has created the need for systems to detect logical access attacks such as speech synthesis and voice conversion to protect the voice-based authentication systems from such malicious attacks.

ASVspoof[1] series started in 2015, and aims to foster the research on countermeasure to detect voice spoofing. In 2015 [1], the challenge focused on detecting commonly used state-of-the-art logical speech synthesis and voice conversion attacks that were largely based on hidden Markov models (HMM), Gaussian mixture models (GMM) and unit selection. Since then, the quality of the speech synthesis and voice conversion systems has drastically improved with the use of deep learning. WaveNet [2], proposed in 2016, was the first end-to-end speech synthesizer that directly uses the raw audio for training, and showed a mean opinion score (MOS) very close to human speech. Similar quality was shown by other TTS systems such as Deep Voice [3] and Tacotron [4], and also by VC systems [5, 6]. These breakthroughs in TTS and VC technologies made the spoofing attacks detection more challenging.

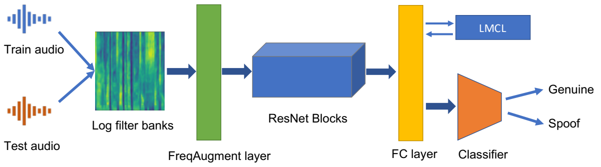


Figure 1: *Overview of the proposed spoofing detection system. 60-dimensional linear filter banks (LFBs) are extracted from raw audio and fed into a Residual Network. FreqAugment layer and large margin cosine loss are used during training the ResNet model. After training this model, same training utterances are fed into ResNet to extract spoofing embeddings, that are then used to train the back-end genuine-vs-spoof classifier.*

In 2019, the ASVspoof [7] logical access (LA) dataset included seventeen different TTS and VC techniques. The organizers took good care of evaluating spoofing detection systems against *unknown* spoofing techniques by excluding eleven *unknown* technologies from train and development datasets. Therefore, strong robustness is required for spoofing detection system in this dataset.

The challenge results show that the current biggest problem in a spoofing detection system is its generalization ability. Traditionally, signal processing researchers tried to overcome this problem by engineering different low-level spectro-temporal features. For example, constant-Q cepstral coefficients (CQCC) were proposed in [8], cosine normalized phase and modified-group delay (MGD) were studied in [9, 10]. Although these works have confirmed the effectiveness of various audio processing techniques in detecting synthetic speech, they are not able to narrow down the generalization gap on ASVspoof 2019 dataset with the recent improved TTS and VC technologies. A detailed analysis of 10 different acoustic features, including linear frequency cepstral coefficient (CQCC) and mel frequency cepstral coefficient (MFCC), was made on ASVspoof 2019 dataset in [11]. The results show that none of these acoustic features are able to generalize well on *unknown* spoofing technologies. Also, using deep learning models to learn discriminate feature embeddings for audio spoofing detection was studied in [12, 13, 14]. A comprehensive study of different tra-

[1] http://www.asvspoof.org

10.21437/Odyssey.2020-19

ditional acoustic features and learned feature from autoencoder was made in [15].

In this work, we tackle this challenge from a different perspective. Instead of investigating different low level audio features, we try to increase the generalization ability of the model itself. To do so, we use large margin cosine loss function (LMCL) [16] which was initially used for face recognition. The goal of LMCL is to maximize the variance between genuine and spoofed class and, at the same time, minimize intra-class variance. Additionally, inspired by *SpecAugment* [17], we propose to add *FreqAugment*, a layer that randomly masks adjacent frequency channels during the DNN training, to further increase the generalization ability of the DNN model. On the ASVspoof 2019 EVAL dataset, we achieve an EER of 1.81% which is significantly better than the baseline. The proposed system is illustrated in Figure 1.

Furthermore, we investigate the effectiveness of audio augmentation techniques. We augment the audio files using publicly available noises, including freely available movies and TV shows, music, other noises and room impulse responses to train and evaluate our system under a noisy scenario. Adding augmented data in the training dataset further reduces the EER from 1.81% to 1.64% on the ASVspoof 2019 EVAL dataset.

Finally, we study the performance of the proposed spoofing detection system in a call center environment. Therefore, we logically-replay the ASVspoof 2019 dataset through VoIP channel to simulate the spoofing attacks. Interestingly, we found that, by adding those audio samples to the training data, the EER is further reduced from 1.64% to 1.26% on the ASVspoof 2019 EVAL dataset.

This paper is organized as follows: Section 2 describes the datasets used to train and evaluate the proposed spoofing detection system. Section 3 details the proposed spoofing detection system. Section 4 presents the experimental results on different evaluation datasets. Section 5 concludes this paper.

2. Datasets

We use three different training protocols and three different evaluation benchmarks as shown in Table 1 and Table 2. The following sections briefly describe the dataset and the data augmentation method used in this work.

2.1. ASVspoof 2019 Challenge Dataset

ASVspoof 2019 [7] logical access (LA) dataset is derived from the VCTK base corpus. It includes seventeen text-to-speech (TTS) and voice conversion (VC) techniques. The spoofing techniques are divided into two groups, six as *known* techniques, eleven as *unknown* techniques. The entire dataset is partitioned into training, development and evaluation sets. The train and development sets include spoofed utterances generated from two *known* voice conversion and four speech synthesis techniques. However, only two *known* techniques are present in the evaluation set. The remaining spoofed utterances were generated from eleven *unknown* algorithms. The training and evaluation parts of this data are named T1 and E1, respectively.

2.2. Augmented ASVspoof 2019

In order to evaluate our system under noisy conditions, data augmentation is performed on original ASVspoof 2019 dataset by modifying the the data augmentation technique from Kaldi. Two types of distortions were used to augment the ASVspoof 2019 dataset: reverberation, and background noise. Room impulse responses (RIR) for reverberation were chosen from publicly available RIR datasets[2] [18, 19, 20]. We chose four different types of background noises for augmentation - music, television, babble, and *freesound*[3]. One part of the background noise files for augmentation were selected from the open source MUSAN noise corpus [21]. We also constructed a television noise dataset using audio segments from publicly available movies and TV shows from Youtube. Around 40 movies and as many TV show videos were downloaded and segmented into 30 second segments to construct the TV-noises set. In all, we collected around 46 hours of TV-noises in our dataset. For music and TV-noises, the audio was reverberated using a randomly selected RIR from the RIR dataset. Then the speech utterances were reverberated using randomly chosen RIRs and then the reverberated noise was added to the reverberated speech utterance. Babble noise was generated by mixing *usgov* utterances from the MUSAN corpus. The *freesound* noises were the general noise files from the MUSAN corpus which consisted of files collected from *freesound* and *soundbible*. For babble and *freesound* noises, we added the background noise files to the clean audio and then reverberated the mixture using a randomly selected RIR. The noises were added with a random SNR between 5dB to 20dB. The training part of this data together with T1 is depicted as T2. Similarly, the evaluation part of this data together with E1 is named E1.

Protocols	Datasets	# Utterances
T1	ASVspoof 2019 train set	25,380
T2	T1 + Augmented train set	152,280
T3	T2 + logically-replayed train set	177,660

Table 1: *The three training protocols used in this work. T1 is the official protocol of the ASVspoof 2019 LA challenge.*

Benchmarks	Datasets	# Utterances
E1	ASVspoof 2019 eval set	71,237
E2	Augmented eval set	356,185
E3	logically-replayed eval set	71,237

Table 2: *The three evaluation benchmarks used to measure the system performance under different acoustic conditions. E1 is the official evaluation set of the ASVspoof 2019 LA challenge.*

2.3. Logically-Replayed ASVspoof 2019

To simulate voice spoofing in a call center environment, Twilio's Voice service[4] is used to playback ASVSpoof 2019 data over voice calls and recorded at the receiver's end. The resulting dataset has VoIP channel characteristics and has reduced bandwidth from 16kHz to 8kHz sampling rate. Twilio's default OPUS codec[5] was used for encoding and decoding audio. This dataset is used to evaluate benchmark (E3) to understand how well our spoofing detection system generalizes in a call-center environment. Also, the replayed training set is added to the protocol (T3). During training and testing, the dataset was upsampled to 16kHz. The training part of this data together with T2

[2] http://www.openslr.org/28/

[3] https://freesound.org/

[4] https://support.twilio.com/hc/en-us/articles/360010317333-Recording-Incoming-Twilio-Voice-Calls

[5] https://www.opus-codec.org/

Layer	Filter size	# filters	Stride	Output size
Freq Masking	-	-	-	200 × 60
Conv1	3 × 3	64	1 × 2	200 × 30
MaxPooling	1 × 3	-	1 × 4	200 × 7
Res block 1	3 × 3	64	1 × 1	200 × 7
Res block 2	3 × 3	128	1 × 1	200 × 4
Res block 3	3 × 3	256	1 × 1	200 × 2
Res block 4	3 × 3	512	1 × 1	200 × 1
Mean and std	-	-	-	1024
FC1	-	-	-	512
FC2	-	-	-	256
LMCL output	-	-	-	2

Table 3: *This table details the architecture of proposed ResNet. All convolutional and fully connect layers are followed by batch normalization and selu activation layer. The outputs from FC2 layer are used as feature embeddings.*

is named T3. Similarly, the evaluation part of this data together with E2 is named E3.

3. Methodology

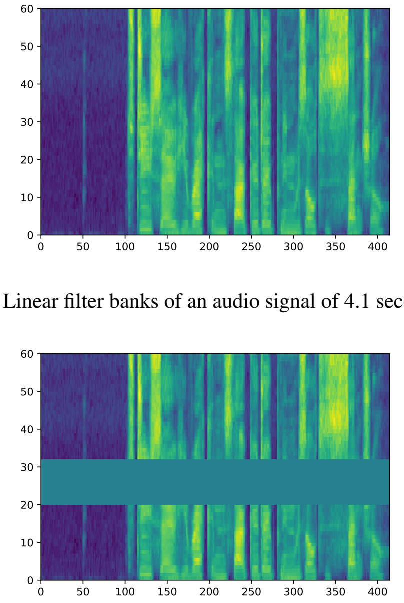
In this section, we first describe the low-level features (Sec 3.1). Then, we introduce the frequency masking layer (Sec 3.2) and large margin cosine loss (Sec 3.3). Next section details the architecture of the embedding extractor (Sec 3.4). Finally, we present the overall spoofing detection system (Sec 3.5).

3.1. Low-level Features

The low-level features used in this work are linear filter banks (LFBs). LFBs are a direct compressed version of the short-time Fourier transforms (SFT), and thus more adequate for lower computational cost. Additionally, they introduce lower risk of network overfitting at training time. Similar filter bank based cepstral features such as linear frequency cepstral coefficients (LFCCs) also showed competitive performance in synthetic speech detection [22] and speaker recognition [23]. We use 60-dimensional LFBs extracted on 30 ms windows with a 10 ms frame shift. Mean and variance normalization was performed on the utterance level. It is worth noting that no voice activity detection is employed.

3.2. Frequency Masking

Online frequency masking is applied during training to randomly drop out a consecutive frequency band range of $[f_0, f_0 + f)$. f is chosen from a uniform distribution $[0, F]$, where F defines the maximum number of frequency channels to be masked. Similarly f_0 is chosen from $[0, v - f]$, where v defines the total number of frequency channels of the input LFB. The f and f_0 are randomly selected, and differ for every mini-bach during training. The same frequency mask is applied on all the training samples within a mini-batch. After creating frequency mask, an element wise multiplication operation is done between original LFB and the frequency mask, so that value of the selected frequency channel can be set to zero. Figure 2 shows an illustration of the LFB after frequency masking. In this work, we set F to 12.



(a) Linear filter banks of an audio signal of 4.1 seconds.

(b) Output of the FreqAugment layer.

Figure 2: *Comparison between the original and masked LFBs. During training, the size and position of the frequency mask are randomly selected, and differ for every mini-batch.*

3.3. Large Margin Cosine Loss

Large margin cosine loss (LMCL) is originally proposed in [16]. It aims to force the deep neural network to learn the feature embedding that can maximize the inter-class variance and minimize the intra-class variance, by reforming the softmax loss as a cosine loss and injecting a margin in the cosine space. LMCL can be defined as:

$$L_{lmc} = \frac{1}{N}\sum_{i} -log\frac{e^{s(cos(\theta_{y_i,i})-m)}}{e^{s(cos(\theta_{y_i,i})-m)} + \sum_{i\neq y_i} e^{scos(\theta_{j,i})}} \tag{1}$$

subject to:

$$\begin{aligned} W &= \frac{W^*}{\|W^*\|}, \\ x &= \frac{x^*}{\|x^*\|}, \\ cos(\theta_j, i) &= {W_j}^T x_i. \end{aligned} \tag{2}$$

where N is the number of training samples, x_i and W_i denote the normalized i-th feature and weight vector corresponding to i-th class. W_j denotes the weight vector of j-th class. s and m are the hyper parameters to define the margin in cosine space. In this work, we set $s = 10$ and $m = 0.35$.

3.4. Deep Residual Network

The **baseline** is our single system submission to the ASVspoof 2019 challenge, which consists in a deep residual network (ResNet) based system. ResNet allows us to train an extensively deeper network to achieve more compelling results. It mitigates the gradient vanishing problem in deep neural networks by stacking residual blocks. The residual block is formed

	Model		
Techniques	Resnet18	Resnet18-L	Resnet18-L-FM
A07	1.10%	0.611%	**0.489%**
A08	1.874%	0.954%	**0.407%**
A09	0.024%	0.017%	**0.017%**
A10	1.589%	1.239%	**0.693%**
A11	0.750%	0.384%	**0.081%**
A12	**0.367%**	0.506%	0.896%
A13	0.611%	0.292%	**0.105%**
A14	4.604%	2.193%	**1.345%**
A15	1.409%	0.873%	**0.628%**
A16	1.891%	1.076%	**0.791%**
A17	12.780%	7.589%	**6.186%**
A18	5.803%	9.504%	**1.548%**
A19	3.643%	3.127%	**2.030%**

Table 4: Detailed performance analysis on different spoofing techniques. All the spoofing techniques listed above were not included in the training protocols. Resnet18-L-FM model has the lowest EER on almost all the spoofing categories.

by adding short-cut connections in between the convolutional layers. This enables the gradients to flow to any other earlier layer. In this work, we use pre-activation residual block proposed in [24]. The Residual Network is a variant of the ResNet-18 described in [25] where the global average pooling (GAP) layer is replaced by mean and standard deviation pooling layers [26]. Additionally, we made some minor changes to the filter size and stride so that the time resolution can be preserved after any convolutional layer and any residual block. After mean and std pooling layers, the concatenated feature map is fed into two consecutive fully connected layers with *scaled exponential linear unit* (selu) activation [27]. We use length normalization layer after the first fully connected layer to further regularize our model. We train the ResNet to classify the audio recordings into two classes: *bonafide* and *spoofed*, and the feature embedding is extracted from the length normalization layer.

The **proposed system** in this paper is an improvement over the baseline, where most of the architecture remain the same. However, we applied frequency masking augmentation before the input layer, remove the length normalization layer, and replace the *softmax* loss with *large margin cosine loss* (LMCL) during training stage. Table 3 shows the detailed implementation and parameters of the proposed model. The ResNet model is trained with ADAM optimizer [28] over 50 epochs.

3.5. System Architecture

Figure 1 shows an overview of our proposed spoofing detection system. First, LFBs are extracted from the raw audio. Then, the LFBs are fed into ResNet embedding extractor to generate deep feature representations. The feature representations are length normalized and fed into the backend classifier to decide whether it is spoofed or genuine audio. The backend classifier is a neural network that consists of one fully connected layer (FC) with 256 neurons, followed by batch normalization layer, dropout layer with dropout rate of 50%, and one softmax output layer.

The ResNet embedding extrator model and the backend classifer are trained separately. After training the embedding extractor model, same training utterances are fed into the embedding extractor to extract feature embeddings, that are then used to train the backend classifier. The embedding extractor and the backend classifier are both trained with ADAM optimizer over 50 epochs.

Model	Training protocol	EER	t-DCF
ResNet18	T1	4.04%	0.109
ResNet18-L	T1	3.49%	0.092
ResNet18-L-FM	T1	**1.81%**	0.052

Table 5: *EER of different spoofing detection systems. L and FM denotes LMCL and frequency masking.*

4. Experiments

This section describes the experimental setup and evaluation metrics, and presents the experimental results.

4.1. Experimental Setup

As described in Section 2, Table 1, and Table 2, we construct three different training recipes and three different evaluation benchmarks. The size of the training sets gradually increase from T1 to T3. Similarly, the difficulty of the evaluation benchmarks gradually increase from E1 to E3. In this paper, the detection task is to verify whether the utterance is genuine (positive class) or not (negative class). The higher the score is, the more likely the utterance is genuine.

Two key performance metrics are used to evaluate the systems. The first is EER that represents the point where false rejection rate (FRR) equals the false acceptance rate (FAR). In this case, the negative class is spoofing. Additionally, we plot the detection error trade-off (DET) curve that shows the accuracy of the our system at different FRRs and FARs.

The second metric is the minimum normalized tandem detection cost function (t-DCF) [29]. The t-DCF is defined as follows:

$$t\text{-}DCF_{norm}^{min} = min\left\{\beta P_{miss}^{cm}(s) + P_{fa}^{cm}(s)\right\} \quad (3)$$

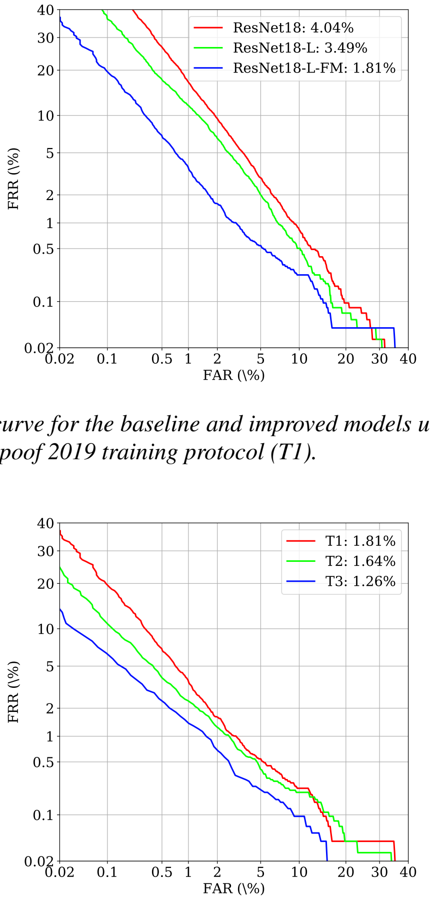
where β depends on application parameters (priors, costs) and ASV performance, $P_{miss}^{cm}(s)$ and $P_{fa}^{cm}(s)$ are the countermeasure system miss and false alarm rate at threshold s. In contrast to the EER computation, the negative class in t-DCF computation is either spoofing or zero-effort impostor. Therefore, the ASV scores should be provided. To keep the comparison fair with ASVspoof 2019 challenge, we used the same ASV scores provided by the organizers.

4.2. Results

Detailed results are shown in Table 5 and Table 6. During the ASVspoof 2019 challenge, our single system submission described in Section 3.4 has achieved an EER of 4.04% and t-DCF of 0.109 on LA evaluation dataset (E1), which is the best performing single system among all submissions. In this paper, we have made a significant improvement over our state-of-the-art system. By replacing the softmax with LMCL, the EER drops to 3.49% and t-DCF 0.092. This shows the LCML is able to force the model to learn more robust features that have better generalization ability. Then, we add frequency masking layer and further reduce the EER to 1.81%. Figure 3(a) plots the different DET curves of the three systems. It clearly shows that the proposed methods bring good improvements over the baseline on most operating points. A detailed performance analysis on detecting different TTS and VC methods are listed in Table 4.

	Benchmarks		
Training Protocols	E1	E2	E3
T1	1.81%	20.43%	8.70%
T2	1.64%	5.34%	8.21%
T3	**1.26%**	**5.32%**	**2.62%**

Table 6: *Performance of ResNet18-L-FM model trained using different protocols.*



(a) *DET curve for the baseline and improved models using the official ASVspoof 2019 training protocol (T1).*

(b) *DET curve for the three different training protocols.*

Figure 3: *DET curves on the ASVspoof 2019 (LA) evaluation benchmark (E1). In these plots, the genuine label is the positive class and the spoofing label is the negative class.*

We further investigate the evaluation and training of our spoofing detection system under noisy and telephony conditions. Table 6 illustrates the detailed results. First, The system trained with T1 achieves an EER of 20.43% on noisy conditions (E2). However, by adding augmented data to the training set (i.e. training on T2), the EER on E2 drops to 5.34%. And more importantly, the EER on E1 dataset is reduced to 1.64%. Finally, we evaluate the proposed system under call center environment (E3). The system achieves reasonable EERs of 8.70% and 8.21% on T1 and T2, respectively. However, by adding telephony data to the training set (i.e. training on T3), the EER on E3 drops to 2.62%. Surprisingly, T3 also improves the overall performance on E1 and E2.

Figure 3(b) plots the different DET curves on E1. They clearly show the effectiveness of doing data augmentation as an important approach towards better generalization.

5. Conclusions

In this paper, we propose a robust end-to-end deep learning framework for voice spoofing detection, that can detect spoofed audio generated from a wide variety of *unknown* TTS and VC systems with high accuracy. We successfully demonstrate that we can increase the generalization ability by adding *FreqAugment* layer and large-margin cosine loss and applying data augmentation. The experimental results show an EER of 1.26% on ASVspoof 2019 evaluation set, which is a remarkable improvement over the state-of-the-art.

6. Acknowledgement

We would like to thank Eric Lorimer, who has helped in collecting the logically-replayed ASVspoof 2019 dataset.

7. References

[1] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[2] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio.," *SSW*, vol. 125, 2016.

[3] Sercan Ö Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al., "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 195–204.

[4] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Zongheng Yang Jaitly, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, et al., "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017.

[5] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.

[6] Tomi Kinnunen, Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, and Zhenhua Ling, "A spoofing benchmark for the 2018 voice conversion challenge: Leveraging from spoofing countermeasures for speech artifact assessment," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 187–194.

[7] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Hector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and

Kong Aik Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[8] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in *Odyssey 2016, The Speaker and Language Recognition Workshop*, 2016.

[9] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," 2012.

[10] Zhizheng Wu, Tomi Kinnunen, Eng Siong Chng, Haizhou Li, and Eliathamby Ambikairajah, "A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case," 2012.

[11] Rohan Kumar Das, Jichen Yang, and Haizhou Li, "Long range acoustic and deep features perspective on asvspoof 2019," in *IEEE Autom. Speech Recognit. Understanding Workshop*, 2019.

[12] Yanmin Qian, Nanxin Chen, Heinrich Dinkel, and Zhizheng Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.

[13] Hossein Zeinali, Themos Stafylakis, Georgia Athanasopoulou, Johan Rohdin, Ioannis Gkinis, Lukáš Burget, Jan Černockỳ, et al., "Detecting spoofing attacks using vgg and sincnet: But-omilia submission to asvspoof 2019 challenge," *arXiv preprint arXiv:1907.12908*, 2019.

[14] Yexin Yang, Hongji Wang, Heinrich Dinkel, Zhengyang Chen, Shuai Wang, Yanmin Qian, and Kai Yu, "The sjtu robust anti-spoofing system for the asvspoof 2019 challenge," *Proc. Interspeech 2019*, pp. 1038–1042, 2019.

[15] Balamurali BT, Kin Wah Edward Lin, Simon Lui, Jer-Ming Chen, and Dorien Herremans, "Towards robust audio spoofing detection: a detailed comparison of traditional and learned features," *arXiv preprint arXiv:1905.12439*, 2019.

[16] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[17] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[18] Satoshi Nakamura, Kazuo Hiyane, Futoshi Asano, and Takashi Endo, "Sound scene data collection in real acoustical environments," *Journal of the Acoustical Society of Japan*, vol. 20, no. 3, pp. 225–231, 1999.

[19] Marco Jeub, Magnus Schafer, and Peter Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[20] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[21] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A music, speech, and noise corpus," *CoRR*, vol. abs/1510.08484, 2015.

[22] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi, "A comparison of features for synthetic speech detection," in *Interspeech*. 2015, ISCA (the International Speech Communication Association).

[23] Xinhui Zhou, Daniel Garcia-Romero, Ramani Duraiswami, Carol Espy-Wilson, and Shihab Shamma, "Linear versus mel frequency cepstral coefficients for speaker recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 559–564.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP*, 2018.

[27] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter, "Self-normalizing neural networks," in *Advances in neural information processing systems*, 2017, pp. 971–980.

[28] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A Reynolds, "t-dcf: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," *arXiv preprint arXiv:1804.09618*, 2018.