

Low Pass Filtering and Bandwidth Extension for Robust Anti-spoofing Countermeasure Against Codec Variabilities

Yikang Wang^{1,2}, Xingming Wang^{2,3}, Hiromitsu Nishizaki¹, Ming Li^{2,3*}

¹Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences,
University of Yamanashi, Japan

²Data Science Research Center, Duke Kunshan University, China

³School of Computer Science, Wuhan University, China

ming.li369@duke.edu

Abstract

A reliable voice anti-spoofing countermeasure system needs to robustly protect automatic speaker verification (ASV) systems in various kinds of spoofing scenarios. However, the performance of countermeasure systems could be degraded by channel effects and codecs. In this paper, we show that using the low-frequency subbands of signals as input can mitigate the negative impact introduced by codecs on the countermeasure systems. To validate this, two types of low-pass filters with different cut-off frequencies are applied to countermeasure systems, and the equal error rate (EER) is reduced by up to 25% relatively. In addition, we propose a deep learning based bandwidth extension approach to further improve the detection accuracy. Recent studies show that the error rate of countermeasure systems increase dramatically when the silence part is removed by Voice Activity Detection (VAD), our experimental results show that the filtering and bandwidth extension approaches are also effective under the codec condition when VAD is applied.

Index Terms: anti-spoofing, bandwidth extension, low-pass filters, band trimming, channel robustness, transmission codec

1. Introduction

Automatic speaker verification (ASV) [1, 2] has achieved good performance and is widely used in real life. As a biometric method, however, ASV systems are vulnerable against various spoofing attacks, e.g. speech synthesis, voice conversion, record and playback, etc. [3, 4]. Anti-spoofing countermeasure systems contribute to enhance the reliability of ASV systems by determining whether the input signal is genuine or spoofed.

Since 2015, the ASVspoof community has initiated and organized four consecutive biennial challenges to support the development of anti-spoofing countermeasure methods for ASV systems [5, 6, 7, 8]. Through this series of challenges, the ASV anti-spoofing field has established two main anti-spoofing countermeasure research scenarios and databases: logical access (LA) considers spoofing attacks from text-to-speech synthesis (TTS) and voice conversion (VC), physical access (PA) refers to attacks produced by recording replay [9, 10]. In this paper, we focus on the LA scenario. A typical countermeasure system consists of a front-end feature extractor and a back-end spoofing classifier. For the LA task, the most intuitive solution is to find the artifact traces existing in the speech from TTS and VC through the front-end operations, which can be used as a marker or cue to distinguish genuine speech from artifacts and help train the back-end detection network better. Previous

works show that artifacts of synthetic speech exist in multiple specific subbands [11, 12]. More and more studies focus on the impact of subbands in countermeasure systems [13].

The current strategies for using frequency subbands in countermeasure systems can be broadly classified into two categories. One uses transformations, trimming, fusion or other methods in the front-end feature extractor to transform the features to a specific domain, which emphasize the information in target subbands [11, 13, 14, 15]. The other one directly adopt the spectrogram as the features, use SpecAugment [16] or similar data augmentation techniques to randomly or specifically block a portion of the frequency bands or time frames to reduce the overfitting of the back-end classifier [17, 18].

Zhang et al. point out that the high-frequency part of the spectrogram may lead to overfitting of the back-end neural network, increasing its risk of making incorrect judgments in the face of unknown spoofing attacks [13]. Moreover, Tomilov et al. find that a data augmentation such as feeding training data into a filter to emulate the magnitude responses of codecs can yield better anti-spoofing countermeasure performance [19].

However, it's not clear that whether low pass filtering is always useful with different bandwidths or scenarios, and whether the re-estimated high frequency information from a deep learning based up-sampling approach could bring some additional gain. This paper investigates the optimal cutoff frequency in terms of low pass filters against the codec variabilities. Moreover, inspired by Tomilov et al's work [19], we use a conventional low-pass filter to obtain a subband signal instead of trimming the spectrogram directly. In addition, we further demonstrate the gain when using a deep learning based bandwidth extension technique to restore the wide band signal from the low pass filtered narrow band speech.

As shown in Figure 1, we investigate the performance of countermeasure systems based on four front-end feature extraction methods and two convolutional neural networks (CNN) back-end classifiers. We also explore the performance of our countermeasure systems after applying a Voice Activity Detection (VAD) module. Our contributions are mainly threefold:

- We investigate the differences when using low-frequency subbands at the system's input features in two different databases and find out that high frequency subbands are more vulnerable against the codecs.
- We use extensive experimental results to validate the optimal cutoff frequency for our countermeasure systems.
- We first utilize a deep learning based bandwidth extension technique on the down-sampled signal in the countermeasure system, and suggest that the additional bandwidth extension module can be effective on the valid speech part when a VAD is applied.

*Corresponding author.

This research is funded in part by the National Natural Science Foundation of China (62171207). Many thanks for the computational resource provided by the Advanced Computing East China Sub-Center.

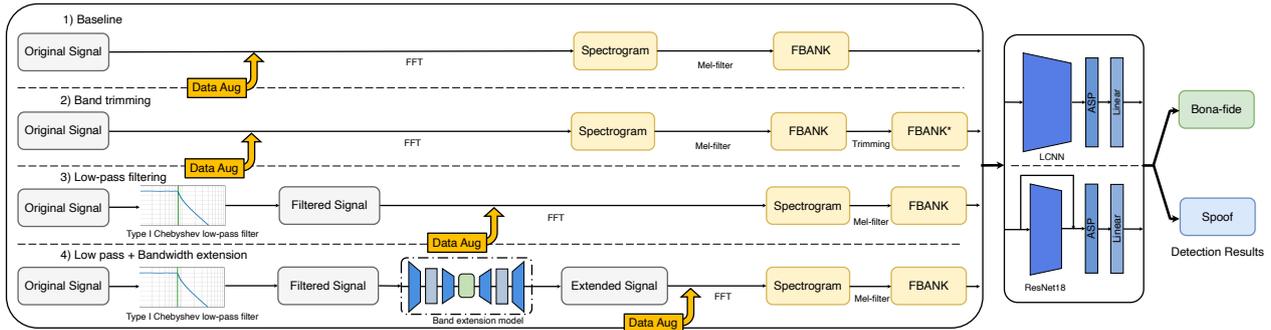


Figure 1: The overview of the proposed front-end signal processing methods.

Table 1: Number of utterances in ASVspooft 2019 and 2021 LA database.

Subset	19LA		21LA	
	Bona-fide	Spoof	Bona-fide	Spoof
Training	2,580	22,800	-	-
Development	2,548	22,296	-	-
Evaluation	7,355	63,882	14,816	166,750

2. Database and Methodology

This section describes the database and illustrates the detail of our front-end signal processing methods shown in Figure 1.

2.1. Databases

The experiments in this work are mainly conducted on the ASVspooft2019 LA database [7] and the ASVspooft2021 LA database [8]. Both databases are based upon the Voice Cloning Toolkit (VCTK) corpus [20]. The ASVspooft2019 LA database was created using utterances from 107 speakers (46 male, 61 female). The set of 107 speakers is partitioned into three speaker-disjoint sets for training, development, and evaluation. The spoofed utterances were generated using four TTS and two VC algorithms in the training and development sets, while 13 TTS/VC algorithms are used in the evaluation subset, 11 of which are unknown for training and development. The ASVspooft2021 LA database remains the training and development data unchanged and only proposes a new evaluation subset that contains attacks using the same simulation methods as the ASVspooft2019 LA evaluation subset. The ASVspooft2021 LA evaluation subset consists of the data in various telephone transmission systems, including Voice over Internet Protocol (VoIP) and the Public Switched Telephone Networks (PSTN), thus exhibiting real-world signal transmission channel effects. These effects could generate artifacts different from spoofing data that affect the accuracy of the countermeasure system. These conditions make the ASVspooft2021 LA evaluation subset an excellent platform for evaluating countermeasure systems' generalization capability and robustness against channel effects and codec variabilities. The contents of the two databases are summarized in Table 1.

2.2. Baseline

We adopt the log Mel-filter bank energy (FBANK) as the acoustic feature in all our experiments. The Fast Fourier Transform (FFT) spectrogram is extracted with 1024 window length and 128 hop length while the Blackman window is used. Then we set the number of Mel-filters to 80 dimensions. Due to the different speech lengths, each spectrogram's length is truncated or

Table 2: The architecture of ResNet18, C denotes the convolutional layer, S denotes the shortcut convolutional layer.

Layer	Output Size	Structure(kernal size, stride)
Conv1	$16 \times D \times L$	$C(3 \times 3, 1)$
Residual Layer 1	$16 \times D \times L$	$\begin{bmatrix} C(3 \times 3, 1) \\ C(3 \times 3, 1) \end{bmatrix} \times 2$
Residual Layer 2	$32 \times \frac{D}{2} \times \frac{L}{2}$	$\begin{bmatrix} C(3 \times 3, 2) \\ C(3 \times 3, 1) \\ S(1 \times 1, 2) \end{bmatrix} \begin{bmatrix} C(3 \times 3, 1) \\ C(3 \times 3, 1) \end{bmatrix} \times 2$
Residual Layer 3	$64 \times \frac{D}{4} \times \frac{L}{4}$	$\begin{bmatrix} C(3 \times 3, 2) \\ C(3 \times 3, 1) \\ S(1 \times 1, 2) \end{bmatrix} \begin{bmatrix} C(3 \times 3, 1) \\ C(3 \times 3, 1) \end{bmatrix} \times 2$
Residual Layer 4	$128 \times \frac{D}{8} \times \frac{L}{8}$	$\begin{bmatrix} C(3 \times 3, 2) \\ C(3 \times 3, 1) \\ S(1 \times 1, 2) \end{bmatrix} \begin{bmatrix} C(3 \times 3, 1) \\ C(3 \times 3, 1) \end{bmatrix} \times 2$
Pooling	$128 \times \frac{D}{8}$	Attentive Statistics Pooling
Linear	128	Fully Connected Layer
Linear	2	Fully Connected Layer

Table 3: The architecture of LCNN.

Layer	Structure(kernal size, stride)	Output Size
Same as the first 28 layers of LCNN [17]		
Pooling	BiLSTM	80 + 80
Pooling	Attentive Statistics Pooling	320
Linear	Fully Connected Layer	128
Linear	Fully Connected Layer	2

concatenated into 3 to 5 seconds. Finally, the $80 * frames$ -dimensional FBANK features are obtained.

For the backend classifier, we investigated two main CNN networks, ResNet18 [21] and LCNN [17]. Then we add the attentive statistics pooling layer (ASP) [22, 2] at the end of the models to make the model more effective in capturing utterance-level acoustic feature changes. The ResNet18 and the LCNN are commonly and widely used systems in anti-spoofing tasks [23]. We use softmax cross entropy as the loss function of the classifier. The architectures of the models are shown in Tables 2 and 3, respectively.

2.3. Band trimming

Band trimming means cropping a particular dimension of the Mel-filter bank from the complete FBANK to make it consistent with the spectrogram subbands. According to the Nyquist frequency characteristic, the spectrogram of the speech with a

sampling rate of 16k covers a bandwidth of 0-8k Hz, and we need to select the low-frequency subband cover 0- \mathbb{F} Hz for training and testing the countermeasure system, which \mathbb{F} here denotes the subband frequencies corresponding to 20%, 30%, 40%, 50%, 60%, 70% Nyquist frequencies. The equation describing the correlation between the number of low- and full-frequency FBANK dimensions [24] is shown as follows

$$\lfloor N_L \rfloor = \lfloor N_F * \frac{\log(1 + f_L/700)}{\log(1 + f_F/700)} \rfloor. \quad (1)$$

Where $\lfloor * \rfloor$ indicates rounding down an element $*$. If the operation of band trimming is considered to be low-pass filtering of the signal, then f_L denotes the cutoff frequency of this filter and f_F refers to the Nyquist frequency. N_L and N_F represent the number of filter banks for the low- and full-frequency FBANK feature, respectively. For instance, when we need to select the low-frequency subband up to 50% of the Nyquist frequency, the corresponding value of f_F , f_L , sample frequency, and N_F are 8,000, 4,000, 16,000, and 80 respectively. Therefore, according to Equation 1, we get $N_L = 60.45$. The feature dimension must be an integer, so we floor N_L to be 60 and set f_L to 3,933.55 Hz. In other words, the spectrogram information from 3,933.55 Hz to 4,000 Hz is dropped out. With respect to Equation 1, 20%, 30%, 40%, 50%, 60%, 70% of Nyquist frequencies corresponds to the FBANK indices 37, 47, 54, 60, 65, and 69.

2.4. Low-pass filtering

Considering the response curve shape, the computational complexity, and the consistency of the filter in the bandwidth extension front-end, the Chebyshev type I filter is chosen as the low-pass filter in our experiments. We set the order of filter to 8, the maximum ripple of the Chebyshev type I low-pass filter to 0.05, and the critical frequencies to \mathbb{F} Hz, which \mathbb{F} is the same as the one mentioned in the last subsection.

2.5. Bandwidth extension

Bandwidth extension is also known as audio upsampling or audio super-resolution. It usually aims to enhance speech audibility and improve audio fidelity by generating a wideband (WB) signal from a narrowband (NB) signal. In order to use extension models to enhance the performance of the countermeasure system in this study, we investigate some bandwidth extension methods [25, 26, 27]. Among them, Viet-Anh Nguyen et al. proposed a Transformer-aided UNet (TUNet)¹ by employing a low-complexity transformer encoder on the bottleneck of a lightweight UNet [28]. Their experimental results on the VCTK corpus show that the TUNet achieves state-of-the-art performance in intrusive and non-intrusive metrics.

The workflow of the bandwidth extension front-end is shown on the left below of Figure 1, where a Chebyshev Type I lowpass filter is used to preprocess the original 16k Hz signal into a low pass filtered signal but still at a sampling rate of 16k Hz. After that, the filtered signal is used as the input of the TUNet extension model [28], and the extended signal containing 0-8k Hz spectrogram can be restored after network inference. Finally, the output of the extension model is changed to FBANK features and as the input of the back-end classifier.

¹Source code: <https://github.com/NXTProduct/TUNet>.

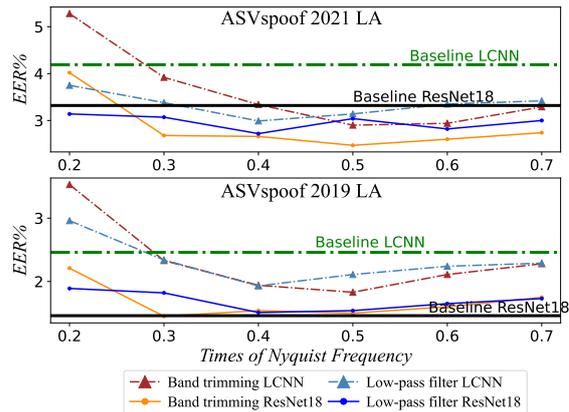


Figure 2: Performance comparison of two subband front-ends (band trimming and low-pass filtering) with the baseline front-end countermeasure system at different cut-off frequencies.

3. Experimental Setup

3.1. Data Augmentation

As shown in Figure 1, in order to improve the robustness of the countermeasure classifier, we implemented data augmentation to add noise before FFT in the front-end. Motivated by the data augmentation methods used in the ASV system with the VoxCeleb database [29, 30, 31], reverberation and background noise are added randomly to two-thirds of the input data. The noise data are obtained from the MUSAN [32] and RIR [33] databases.

3.2. Metric and training strategy

The evaluation was performed in terms of Equal Error Rate (EER) as a metric, which indicates that the proportion of false acceptances is equal to the proportion of false rejections.

For training the countermeasure systems, Adam [34] was used as the optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decaying 10^{-4} . For the CNN classifiers, the learning rate increase linearly for the first four warm-up epochs and then is initialized to 0.001 starting from the fifth epoch. The learning rate is scheduled as the PyTorch ReduceLRonPlateau function², reducing the learning rate when the metric has stopped improving for ten epochs. Each experiment uses one NVIDIA RTX A6000 GPU, and for efficiency, we set the batch to 400, with 100 epochs of training per model.

4. Experimental Results and Discussion

4.1. Countermeasure systems with different front-end

From Figure 2, it seems a cutoff frequency of 0.5 is a good candidate for comparison experiments. The experimental results of all eight countermeasure systems are shown in Table 4, in which the cutoff frequencies were set to 0.5 times the Nyquist frequency. In addition, since some studies have shown that the performance difference caused by different random seeds may even be more significant than that caused by different countermeasure system constructions [35], we repeat the experiment three times for each case of the system using three random seeds, then we calculate the average EER. Table 4 suggests that at 0.5 times the Nyquist frequency, the countermeasure systems based on

²https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLRonPlateau.html

Table 4: The EER% of countermeasure systems, when choosing 0.5 Nyquist frequency as the cutoff frequency. **Bold number** indicates the best performing result among the countermeasure systems corresponding to each back-end network in the 19LA or 21LA database.

back-ends	front-ends	ASVspoof2021 LA				ASVspoof2019 LA			
		seed1	seed10	seed100	Average	seed1	seed10	seed100	Average
ResNet18	baseline	3.23	3.35	3.38	3.32	1.76	1.48	1.15	1.46
	band trimming	2.65	2.32	2.44	2.47	1.37	1.66	1.46	1.5
	low-pass filtering	3.05	3.03	3.05	3.04	1.34	1.7	1.57	1.53
	low pass + bandwidth extension	2.35	2.57	2.72	2.54	1.37	1.38	1.6	1.45
LCNN	baseline	3.95	4.37	3.38	4.26	2.36	2.36	2.5	2.41
	band trimming	3.23	2.72	2.76	2.90	2.03	1.75	1.72	1.83
	low-pass filtering	3.51	2.84	3.07	3.14	2.31	1.89	2.14	2.11
	low pass + bandwidth extension	2.94	2.97	3.01	2.91	2.05	2.05	1.84	1.98

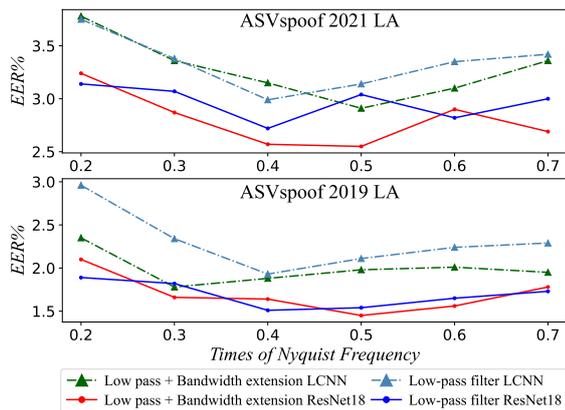


Figure 3: Performance comparison of low-pass filtering and bandwidth extension front-end countermeasure systems.

Table 5: The EER% of 4 different front-ends and ResNet18 back-end countermeasure systems after VAD operation.

back-ends	front-ends	21LA	19LA
ResNet18	baseline	20.17	13.06
	band trimming	19.91	18.24
	low-pass filtering	19.08	16.64
	low pass + bandwidth extension	15.23	15.08

band trimming and bandwidth extension front-ends have similar performance on the ASVspoof2021 LA database. However, the best performance on the ASVspoof2019 LA database is achieved by the baseline front-end system.

Under the six cutoff frequencies specified in this experiment, Figure 2 provides the results of the countermeasure system based on band trimming and conventional low-pass filter front-end as well as the baselines. It can be found that all countermeasure systems composed of low-frequency subband front-ends outperform the baseline system in the ASVspoof2021 LA database when the cutoff frequency is greater than 0.3 times the Nyquist frequency. As this observation is consistent on two baseline systems, we think it because low-frequency subbands can reduce the variabilities of channel effects and codecs as human ears are less sensitive on high frequency regions which might be more distorted in transmission. The results on the ASVspoof2019 LA database show that low-frequency subbands improve the LCNN backend systems performance significantly more than the ResNet18 system. Combined with the conclusion of Zhang et al. [13], we suggest that the low pass filtering or down-sampling to narrow band spectrograms is effective

when there is relative large channel effects or codec variabilities, however, it might not be useful when there is little high frequency variability on speech. In that case, dropping out high frequency information would result in degraded performance. Figure 3 compares the system’s performance before and after the bandwidth extension module. It can be found that the bandwidth extension operation on the low pass filtered signal improves performance in most cases. Combining Figure 2 and Figure 3, we determine the optimal cutoff frequency for band trimming/filtering/bandwidth extension as 0.5/0.4/0.5 times Nyquist frequency.

4.2. The effect of VAD operation

Some studies have shown that since the countermeasure system focuses on silent segments to distinguish spoofed speech from genuine speech, the data after VAD processing will be difficult to classify [13, 19]. We want to further test our proposed methods on the silence removed speech signals after VAD.

We use the `librosa.effects.trim` function³ of the open source toolkit Librosa to implement the VAD function. We set the `top_db` parameter to 40, which means that the part of each data with a maximum energy below 40 dB is considered as silence. The performance of each system is shown in Table 5 with their own optimal cutoff frequency condition, and it can be found that the EER of all systems increased substantially compared to the ones without VAD. The model with the baseline front-end increase the least on the 2019 database, but the most on the 2021 database; while the EER of systems with the bandwidth extension front-end increased less in both databases. It suggests that bandwidth extension can improve the system performance on speech part of the signal significantly. It also suggest that the original silence signal contains large portion of spoofing artifacts, applying additional signal processing methods on it might distort the silence part artifacts.

5. Conclusions

This work validates the countermeasure systems on the ASVspoof2019 LA and the ASVspoof2021 LA databases and shows that low-frequency narrow band can reduce the disturbance caused by channel effects and codec variabilities. In addition, bandwidth extension can significantly reduce the performance degradation after VAD. Moreover, we compared different front-ends and determined the optimal cut-off frequency for those systems. Our future work will focus on exploring the cases that low pass filtering and bandwidth extension are only applied on the speech part of the signal and leave the silence part unchanged.

³<http://librosa.org/doc/latest/generated/librosa.effects.trim.html>

6. References

- [1] 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.
- [2] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.
- [3] M. R. Kamble, H. B. Sailor, H. A. Patil, and H. Li, "Advances in anti-spoofing: from the perspective of asvspoof challenges," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.
- [4] N. Evans, T. Kinnunen, J. Yamagishi, Z. Wu, F. Alegre, and P. D. Leon, "Speaker recognition anti-spoofing," in *Handbook of biometric anti-spoofing*, 2014, pp. 125–146.
- [5] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. HaniŇi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech 2015*, 2015, pp. 2037–2041.
- [6] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in *Proc. Interspeech 2017*, 2017, pp. 2–6.
- [7] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *Proc. Interspeech 2019*, 2019, pp. 1008–1012.
- [8] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *Proc. of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 47–54.
- [9] F. Tom, M. Jain, and P. Dey, "End-To-End Audio Replay Attack Detection Using Deep Convolutional Networks with Attention," in *Proc. Interspeech 2018*, 2018, pp. 681–685.
- [10] R. Baumann, K. M. Malik, A. Javed, A. Ball, B. Kujawa, and H. Malik, "Voice spoofing detection corpus for single and multi-order audio replays," *Computer Speech & Language*, vol. 65, p. 101132, 2021.
- [11] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah, "Investigation of Sub-Band Discriminative Information Between Spoofed and Genuine Speech," in *Proc. Interspeech 2016*, 2016, pp. 1710–1714.
- [12] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2160–2170, 2019.
- [13] Y. Zhang, W. Wang, and P. Zhang, "The Effect of Silence and Dual-Band Fusion in Anti-Spoofing System," in *Proc. Interspeech 2021*, 2021, pp. 4279–4283.
- [14] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "Spoofing Attack Detection Using the Non-Linear Fusion of Sub-Band Classifiers," in *Proc. Interspeech 2020*, 2020, pp. 1106–1110.
- [15] B. Chettri, T. Kinnunen, and E. Benetos, "Subband Modeling for Spoofing Detection in Automatic Speaker Verification," in *Proc. Odyssey 2020*, 2020, pp. 341–348.
- [16] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.
- [17] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 1033–1037.
- [18] R. Yan, C. Wen, S. Zhou, T. Guo, W. Zou, and X. Li, "Audio deepfake detection system with neural stitching for add 2022," in *Proc. ICASSIP 2022*, 2022, pp. 9226–9230.
- [19] A. Tomilov, A. Svishchev, M. Volkova, A. Chirkovskiy, A. Kondratev, and G. Lavrentyeva, "STC Antispoofing Systems for the ASVspoof2021 Challenge," in *Proc. of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 61–67.
- [20] J. Yamagishi, C. Veaux, and K. MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR 2016*, 2016, pp. 770–778.
- [22] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256.
- [23] X. Wang, X. Qin, T. Zhu, C. Wang, S. Zhang, and M. Li, "The DKU-CMRI System for the ASVspoof 2021 Challenge: Vocoder based Replay Channel Response Estimation," in *Proc. of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 16–21.
- [24] W. Cai and M. Li, "A unified deep speaker embedding framework for mixed-bandwidth speech data," in *Proc. APSIPA ASC 2021*, IEEE, 2021, pp. 1133–1138.
- [25] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, no. 8, pp. 1707–1719, 2003.
- [26] K. Zhang, Y. Ren, C. Xu, and Z. Zhao, "WSRGlow: A Glow-Based Waveform Generative Model for Audio Super-Resolution," in *Proc. Interspeech 2021*, 2021, pp. 1649–1653.
- [27] J. Lee and S. Han, "NU-Wave: A Diffusion Probabilistic Model for Neural Audio Upsampling," in *Proc. Interspeech 2021*, 2021, pp. 1634–1638.
- [28] V.-A. Nguyen, A. H. Nguyen, and A. W. Khong, "Tunet: A block-online bandwidth extension model based on transformers and self-supervised pretraining," in *Proc. ICASSP 2022*, 2022, pp. 161–165.
- [29] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In Defence of Metric Learning for Speaker Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2977–2981.
- [30] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.
- [31] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.
- [32] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.
- [33] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP 2017*, 2017, pp. 5220–5224.
- [34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR 2015*, 2015.
- [35] X. Wang and J. Yamagishi, "A Comparative Study on Recent Neural Spoofing Countermeasures for Synthetic Speech Detection," in *Proc. Interspeech 2021*, 2021, pp. 4259–4263.