INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Compressed Synthetic Speech Detection Method with Compression Feature Embedding

Jinghong Zhang[1,2], *Xiaowei Yi*[1,2], *Xianfeng Zhao*[1,2(✉)]

[1]The State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100085, China
{zhangjinghong, yixiaowei, zhaoxianfeng}@iie.ac.cn

Abstract

With the development of deep fake technology, synthetic speech is created easier by forgery techniques based on text-to-speech and voice conversion, which poses a challenge to automatic speaker verification systems. Existing methods demonstrate excellent performance on public databases, but most methods are weak in detecting compressed speech commonly used in social networks, such as MP3 and AAC. We believe that if the classifier has compressed information as a priori knowledge, it will help the classifier make a more accurate decision when detecting compressed speech. To solve this issue, a multi-branch residual network with a compression feature embedding module is proposed in this paper. The feature embedding module is used to integrate the authenticity feature and compression feature. Our method is evaluated on the ASVspoof database and experimental results show the effectiveness of the proposed method for detecting compressed speech.

Index Terms: synthetic speech detection, feature embedding, compression, robustness, ASVspoof

1. Introduction

With the evolution of deepfake technology, it is much easier to generate highly realistic fake human voices than before. The mainstream speech synthesis methods are mainly divided into two forms. One is to synthesize the target character's voice through a paragraph of text, which is called text-to-speech (TTS). Another method is to convert the voice of the source speaker into the voice of the target speaker and keep the content of the voice unchanged, which is called voice conversion (VC). The application of high-performance neural networks makes the synthetic speech and converted speech closer to natural speech in both auditory and feature, which brings greater challenges to speech detection tasks.

In recent years, to address the threats caused by synthetic audio systems, researchers devote themselves to improving the performance of fake audio detection methods by introducing deep learning technology. At first, the neural network is implemented to extract high-dimensional features based on hand-crafted features such as linear frequency cepstral coefficients (LFCC) [1, 2], Mel-frequency cepstral coefficients (MFCC) [3–5], and constant Q cepstral coefficients (CQCC) [6–8]. The neural network also plays the role of back-end classifier at the same time. A mass of classical architectures of neural networks have been applied for detecting synthetic speech, such as residual network (ResNet) [9, 10], squeezed-and-excitation network (SENet) [2, 7] and recurrent neural network (RNN). Gomez-Alanis *et al.* [11] present a light gated convolution network by combining the light convolution neural network (LCNN) and recurrent neural network (RNN) to extract deep features for spoof speech detection. Lai *et al.* [7] propose a detection system named ASSERT, which combines the SENet and ResNet for anti-spoofing. In this system, authors use two different features (CQCC and Logarithmic Spectrum) and various network variants to improve the detection performance, and the detection performance is greatly improved on the ASVspoof2019 dataset. With the development of detection technology, some researchers attempt to use end-to-end neural networks to distinguish real speech from synthetic speech [12, 13]. The end-to-end network uses original audio waveform as input data instead of hand-crafted features to avoid information loss caused by manual intervention in the process of feature extraction.

However, these above methods pay more attention to identifying the authenticity of raw speech files, the detection performance will decrease rapidly when they deal with the compressed speech files. In practice, compressed audio is widely used in our daily life, The audio compressed under MP3 and AAC are the most common audio formats in the social network. The ASVspoof2021 challenge [14] set the DF corpus whose audio is collected by several different compress methods, and most anti-spoof countermeasures tend to perform poorly in this task.

In this work, we propose a residual network with feature embedding architecture for detecting fake speech under compressed conditions. The method aims to evaluate the connection between the authenticity feature and the compression feature. We implement data augment to expand our training set so that we can get more information with compressed audio. The network contains two different branches, namely the authenticity branch and the compression branch. The function of the authenticity branch is similar to that of the traditional detection network, and the output is fake or real. The compression branch aims to distinguish between different compression modes and compression qualities, and the output is multiple labels. The main contributions of our work are summarized as follows:

Compressed fake speech detection method. We propose an adaptive classification method named CFE-ResNet based on residual network and feature embedding module for detecting compressed fake speech. The feature embedding module encodes the characteristic of compressed audio into the feature vector and cat the compressed feature vector with the feature which represents authenticity. Our method can be generalized to detect deep fake speech in various compression conditions.

Data augment. To improve the performance in resisting compression attacks, we use multiple different compression methods and parameters to expand our training set. It includes three different compression formats and two different compression qualities.

Detecting fake speech on ASVspoof2019 and ASVspoof2021. We use fake and real speech from the

10.21437/Interspeech.2023-1696

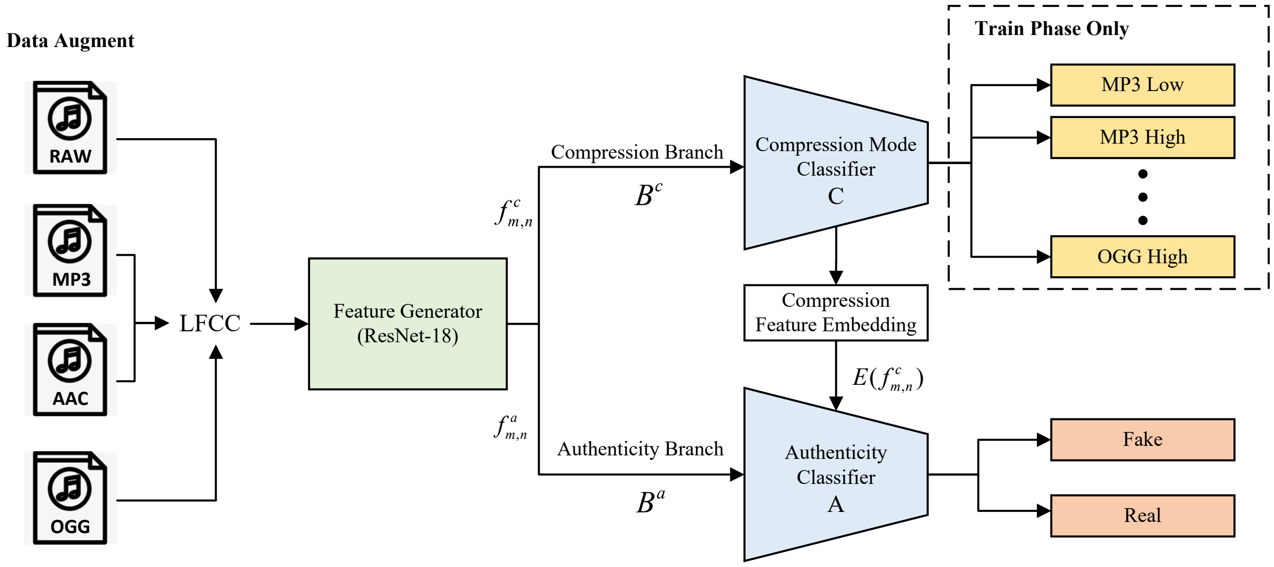

Figure 1: *The structure of CFE-ResNet. The upper branch is the compression branch, and the lower branch is the authenticity branch. The part in the rectangle marked "Train Phase Only" is only used during training, and will be deleted in the evaluation phase.*

ASVspoof2019 logical access (LA) [15] and ASVspoof2021 deep fake (DF) [14] to evaluate our method. The ASVspoof2019 LA dataset is preprocessed by different compression modes. The experimental result shows that our method outperforms other state-of-the-art single system methods in detecting compressed fake speech.

The remaining part of the paper is organized as follows. The architecture of our scheme is presented in Section 2. Experimental datasets and evaluation measures are described in Section 3. Then the performances of the proposed method on the ASVspoof2019 and ASVspoof2021 datasets are described in Section 4. Finally, Section 5 draws conclusions and directions for future work.

2. Proposed method

In this section, we will introduce the structure of ResNet with compression feature embedding (CFE-ResNet) in detail first. Then we will present the front-end feature of our method, the specific structure of our method, and the objective function in the remainder of this section.

2.1. The front-end feature and data augment

In spoofing audio detection, many hand-crafted features can be chosen as front-end features, such as MFCC, LFCC, CQT, and CQCC. The study in [16] shows that the performance of LFCC features is generally better than MFCC features. So in this work, we take LFCC as the front-end feature of our method. The specific parameters of extracting LFCC will be introduced in the experimental section.

To counter the compression attacks, we extend the training dataset via three different compression codecs, namely MP3, AAC, and OGG. And in each compression mode, we set two different compression qualities, namely high quality and low quality, which correspond to high and low bitrates in sequence.

2.2. The structure of CFE-ResNet

2.2.1. Overall network structure

The CFE-ResNet is designed for detecting compressed fake speech via the compression feature embedding module. The main flowchart of CFE-ResNet is shown in Figure 1. As the figure shows, The CFE-ResNet consists of two significant branches: authenticity detection branch D_a and compression detection branch D_c. Before the two branches, a feature generator is set to represent the LFCC feature into the deep feature. In this work, we choose the ResNet-18, which removes the full connection layer, as the feature generator. The two classifiers A, C, and the compression feature embedding module are built by the full connection layer.

Let us assume the training dataset is $D = \{X_{m,n}, y^a_m, y^c_n\}$, where $m = \{0, 1, 2, \dots, M-1\}$ denotes the index of the training sample (M is the total number of Samples) and $n = \{0, 1, 2, \dots, N-1\}$ denotes the index of different augment mode. y^a and y^c represent the label of authenticity and compression mode respectively. In this work, the authenticity label is fake or real. And the compression labels are the combination of four codecs (RAW, MP3, AAC, OGG) and two compression quality (high and low), which are seven different labels in total (RAW does not have compression quality).

The front-end feature $x_{m,n}$ is encoded via the feature generator into latent feature $f_{m,n}$. Later, the $f_{m,n}$ is divided into two parts $f^a_{m,n}$ and $f^c_{m,n}$, which will be sent to different branches. The first branch B^c includes a compression mode classifier C, which tries to extract the feature for representing the different compression modes. The output of B^c branch consists of two parts: the softmax output and the feature embedding output. The function of softmax output is to be calculated loss with the valid label y^c_n to make classifier C correctly distinguish different compression modes. And the purpose of the other output is to add compression information to classifier A. The branch B^a aims to distinguish nature speech and fake speech, which includes an authenticity classifier A. This branch only has one output to predict the presumptive label.

2.2.2. The compression feature embedding

After digital media files are compressed, much redundant information and detailed information will be deleted, resulting in great changes in the feature distribution of the original samples. For statistical learning classification algorithms, especially deep learning algorithms, the detection performance will be greatly reduced. In previous studies, Zhang *et al.* [17] tried to force the classification network to learn the same features under different compression environments by using adversarial decoupling and similarity decoupling modules, to achieve the purpose of defending against compression attacks. In this paper, we consider that compression attacks have a great impact on the distribution of features, especially in low-quality compression conditions, it is very difficult to force the network to learn the same features in different compression conditions. Based on that, we propose a new strategy, giving the network certain information about compression modes to distinguish the authenticity of the samples in a given condition. Before the authenticity feature $f^a_{m,n}$ is put into the back-end classifier, we cat it with the compression feature $f^c_{m,n}$ embedded from the first branch B^c, which is shown in Eq. (1). Then $f^b_{m,n}$ is sent to the downstream classifier A to predict the authenticity label.

$$f^b_{m,n} = f^a_{m,n} \oplus E(f^c_{m,n}) \tag{1}$$

It is worth mentioning that considering the unknown compression conditions, we choose the network instead of the one-hot encoder to encode different compression codecs and qualities.

2.3. The objective function of the proposed method

As mentioned above, the loss function can be divided into two parts: One is the multi-class classification task loss from the branch B^c and the other one is the binary classification task. We apply the standard cross-entropy loss for these branches, which is shown in Eq. (2) and Eq. (3):

$$L_C = -\sum_{m=1}^{M}\sum_{n=1}^{N} y^c_n * \log(\hat{y^c}_{m,n}) \tag{2}$$

$$L_A = -\sum_{m=1}^{M}\sum_{n=1}^{N} y^a_m * \log(\hat{y^a}_{m,n}) \tag{3}$$

where $\hat{y^c}_{m,n} = C(f^c_{m,n})$ and $\hat{y^a}_{m,n} = A(f^b_{m,n})$ represent the predicted label of compression mode and authenticity respectively. The ultimate loss function is to combine the two parts loss, which is shown in Eq. (4)

$$L = L_A + L_C \tag{4}$$

3. Experimental setup

In this section, we introduce the database and metric we used in this work. Then our experimental setup and the training process are described in detail.

3.1. Database and metrics

We choose the ASVspoof2019 database [15] and ASVspoof2021 database [14] to verify the performance of our methods. Table 1 presents a summary of the ASVspoof2019 logical access corpus and the ASVspoof2021 deepfake corpus.

ASVspoof2021 DF corpus includes bonafide and spoofed speech utterances processed with different lossy codecs used typically for media storage. The audio file of DF is uncompressed audio which is decoded from the compressed version. Evaluation source data is taken from the ASVspoof2019 LA evaluation set and Voice Conversion Challenge (VCC) [18, 19]. The DF corpus is generated with more than 100 different spoofing algorithms and nine different codec modes.

Table 1: *Summary of the ASVspoof2019 LA corpus and ASVspoof2021 DF corpus.*

ASVspoof2019	Bonafide		Spoofed	
	#speaker	#utterance	#speaker	#utterance
Training	20	2,580	20	22,800
Development	10	2,548	10	22,296
Evaluation	68	7,533	68	63,882
ASVspoof2021	Bonafide		Spoofed	
	#speaker	#utterance	#speaker	#utterance
Evaluation	93	22,617	63	589,212

ASVspoof2019 LA scenario is derived from the VCTK base corpus which includes speech data captured from 107 speakers (46 males, 61 females). In this work, we make a compressed dataset named ASVspoof2019 deepfake (DF) corpus based on the evaluation set of the LA scenario. The setting of compression modes is the same as ASVspoof2021 DF corpus.

The primary metric in this paper is the equal error rate (EER). EER is defined as the value where the false alarm rate is equal to the miss rate by setting a threshold. The lower EER indicates that the detection method has better detection performance.

3.2. Training details

The training data is preprocessed by the data augment method mentioned In Section 2. We set a 20ms frame size and 10ms hop size to extract the 60-dimensional LFCC feature. To ensure the consistency of data dimensions, we use the settings in [23] to fix the length to 750 frames. For shorter speech, repeat padding is applied to extend the length until longer than the set length. Then we select 750 continuous frames randomly.

Our method is constructed with PyTorch Toolk and trained on a single NVIDIA V100 GPU. In terms of setting up the optimizer, We use the Adam [25] optimizer with the β_1 parameter set to 0.9 and the β_2 parameter set to 0.999 to update the weights in our method. The batch size is set to 64. The learning rate is initially set to 0.001.

4. Result and discussion

In this section, we report the results of the proposed method and analyze the influence of different structures. And we compare the detection performance with other state-of-the-art single system classifiers.

4.1. Ablation study

To demonstrate the effectiveness of the compression branch and compression feature embedding in the proposed method, we set the ablation experiment by implementing the CFE-ResNet with its three variants: i) The CFE-ResNet only reserves the authenticity branch, which is denoted as model 1, ii) The CFE-ResNet reserves the authenticity branch and compression branch but delete the compression feature embedding module, which is denoted as model 2, iii) The CFE-ResNet with compression

Table 2: *EER (%) of ablation experiment with the variant of CFE-ResNet on ASVspoof2019 DF corpus and ASVSpoof2021 DF corpus. "CB" is short for compression branch and "FE" is short for compression feature embedding moudule. "✔" indicates the corresponding structure is used while "✖" indicates not. Bold numbers indicate the best result in the column.*

Method	CB	FE	ASVspoof2019	ASVspoof2021
1	✖	✖	6.14	20.83
2	✔	✖	5.82	18.45
3	✔	✔	**5.23**	**16.20**

Table 3: *EER (%) of comparison with other advanced method on ASVspoof2021 DF corpus. Bold numbers indicate the best result in the column*

Mehtod	ASVspoof2021									
	DF-C1	DF-C2	DF-C3	DF-C4	DF-C5	DF-C6	DF-C7	DF-C8	DF-C9	Pooled
CQCC + GMM [20]	15.84	47.84	18.75	18.59	16.7	20.23	15.87	47.16	47.16	24.58
LFCC + GMM [21]	13.73	41.14	16.67	20.83	19.29	24.72	15.21	37.32	27.71	24.93
LFCC + LCNN [22]	16.55	30.82	20.09	19.52	20.93	20.99	20.34	25.89	21.34	21.68
RawNet2 [13]	19.36	22.92	21.53	20.86	22.29	19.48	20.66	19.13	20.42	20.85
Inc-TSSDNet [12]	22.89	25.3	24.98	27.17	26.56	25.1	29.13	29.08	26.68	26.07
LFCC + OC-softmax [23]	14.43	29.57	40.92	21.8	23.6	25.84	23.25	31.16	24.84	30.95
ECAPA-TDNN [24]	22.04	29.56	24.76	22.86	23.04	16.04	15.74	19.54	**16.17**	20.33
CFE-ResNet (proposed)	**11.23**	**19.65**	**15.58**	**13.5**	**12.46**	**15.36**	**14.39**	**18.08**	16.20	**16.20**

branch and compression feature embedding module. The results are shown in Table 2. From the results, we can see that the performance of method 1 to method 3 is gradually increasing, which can prove that the compression branch and compression embedding module are really useful for detecting compressed audio.

4.2. Comparison with other methods

In this part, we compare our method with other fake speech detection methods. Firstly, we test the detection performance of different methods on the ASVspoof2021 DF corpus, and the results are shown in Table 3. In this experiment, we compare our proposed method with four baseline methods selected by the ASVspoof2021 challenge and three state-of-the-art methods, where ECAPC-TDNN [24] is specially designed for detecting compressed speech. From the table, we can see that our method shows the best performance when dealing with different compression modes (From C1 to C9). The compression modes C8 and C9 use two different ways to compress speech twice, the detection results for these two patterns show that our method is more general than other methods when it comes to unknown compression modes. Then we implement our method and baseline methods on ASVspoof2019 DF corpus described in Section 3 and Table 4 shows the results. The result of this experiment shows that our proposed method shows better performance in resisting compression attacks in evaluation but loses in the development set, which means the end-to-end network has good performance in coping with compression attacks but it lacks certain generalizations in dealing with different data. It is noteworthy that the results in Table 4 are obviously better than that in Table 3 because the data in ASVspoof2021 DF dataset is made from different datasets, which makes the detection task more challenging.

Table 4: *EER (%) of comparison with other advanced method on ASVspoof2019 DF corpus. Bold numbers indicate the best result in the column.*

Mehtod	ASVspoof2019	
	dev set	eval set
CQCC + GMM [20]	12.94	13.3
LFCC + GMM [21]	9.8	16.79
LFCC + LCNN [22]	2.93	9.76
RawNet2 [13]	**0.68**	6.45
Inc-TSSDNet [12]	1.17	5.89
LFCC + OC-softmax [23]	17.14	16.09
CFE-ResNet (proposed)	2.78	**5.23**

5. Conclusion

In this paper, we propose a novel ResNet with compression feature embedding (CFE-ResNet) for detecting compressed fake speech. The most significant structure of the proposed method is the compression feature module, which can extend the knowledge learned from the compression rate detection branch to the authenticity detection branch. The ablation study shows the compression mode classification and compression embedding module can effectively improve the detection performance for detecting compressed speech. Then the experiments on ASVspoof2019 and ASVspoof2021 datasets demonstrate that our proposed outperforms other state-of-the-art single system methods for defending against compression attacks. In the future, we would aim to optimize the structure of our method to improve detection accuracy.

6. Acknowledgement

This work was supported by National Key Technology Research and Development Program under 2020AAA0140000.

7. References

[1] Xing Fan and John HL Hansen. Speaker identification with whispered speech based on modified lfcc parameters and feature mapping. In *2009 IEEE international conference on acoustics, speech and signal processing*, pages 4553–4556. IEEE, 2009.

[2] Xu Li, Na Li, Chao Weng, Xunying Liu, Dan Su, Dong Yu, and Helen Meng. Replay and synthetic speech detection with res2net architecture. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6354–6358. IEEE, 2021.

[3] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu. Resnet and model fusion for automatic spoofing detection. In *Interspeech*, pages 102–106, 2017.

[4] Mohammed Lataifeh, Ashraf Elnagar, Ismail Shahin, and Ali Bou Nassif. Arabic audio clips: Identification and discrimination of authentic cantillations from imitations. *Neurocomputing*, 418:162–177, 2020.

[5] Arun Kumar Singh and Priyanka Singh. Detection of ai-synthesized speech using cepstral & bispectral statistics. In *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 412–417. IEEE, 2021.

[6] Akash Chintha, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. Recurrent convolutional structures for audio spoof and video deepfake detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1024–1037, 2020.

[7] Cheng-I Lai, Nanxin Chen, Jesús Villalba, and Najim Dehak. Assert: Anti-spoofing with squeeze-excitation and residual networks. *arXiv preprint arXiv:1904.01120*, 2019.

[8] Zhenchun Lei, Yingen Yang, Changhong Liu, and Jihua Ye. Siamese convolutional neural network using gaussian probability feature for spoofing speech detection. In *INTERSPEECH*, pages 1116–1120, 2020.

[9] Moustafa Alzantot, Ziqi Wang, and Mani B Srivastava. Deep residual neural networks for audio spoofing detection. *arXiv preprint arXiv:1907.00501*, 2019.

[10] Prasanth Parasu, Julien Epps, Kaavya Sriskandaraja, and Gajan Suthokumar. Investigating light-resnet architecture for spoofing detection under mismatched conditions. In *INTERSPEECH*, pages 1111–1115, 2020.

[11] Alejandro Gomez-Alanis, Antonio M Peinado, Jose A Gonzalez, and Angel M Gomez. A light convolutional gru-rnn deep feature extractor for asv spoofing detection. In *Proc. Interspeech*, volume 2019, pages 1068–1072, 2019.

[12] Guang Hua, Andrew Beng Jin Teoh, and Haijian Zhang. Towards end-to-end synthetic speech detection. *IEEE Signal Processing Letters*, 28:1265–1269, 2021.

[13] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6369–6373. IEEE, 2021.

[14] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. *arXiv preprint arXiv:2109.00537*, 2021.

[15] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor 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.

[16] Massimiliano Todisco, Héctor Delgado, Kong Aik Lee, Md Sahidullah, Nicholas Evans, Tomi Kinnunen, and Junichi Yamagishi. Integrated presentation attack detection and automatic speaker verification: Common features and gaussian back-end fusion. In *Interspeech 2018-19th Annual Conference of the International Speech Communication Association*. ISCA, 2018.

[17] Jian Zhang, Jiangqun Ni, and Hao Xie. Deepfake videos detection using self-supervised decoupling network. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[18] 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. *arXiv preprint arXiv:1804.04262*, 2018.

[19] Yi Zhao, Wen-Chin Huang, Xiaohai Tian, Junichi Yamagishi, Rohan Kumar Das, Tomi Kinnunen, Zhenhua Ling, and Tomoki Toda. Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion. *arXiv preprint arXiv:2008.12527*, 2020.

[20] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516–535, 2017.

[21] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. Spoofing attack detection using the non-linear fusion of sub-band classifiers. *arXiv preprint arXiv:2005.10393*, 2020.

[22] Xin Wang and Junich Yamagishi. A comparative study on recent neural spoofing countermeasures for synthetic speech detection. *arXiv preprint arXiv:2103.11326*, 2021.

[23] You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941, 2021.

[24] Xinhui Chen, You Zhang, Ge Zhu, and Zhiyao Duan. UR Channel-Robust Synthetic Speech Detection System for ASVspoof 2021. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 75–82, 2021.

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