INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Improving Replay Detection System with Channel Consistency DenseNeXt for the ASVspoof 2019 Challenge

Chao Zhang, Junjie Cheng, Yanmei Gu, Huacan Wang, Jun Ma, Shaojun Wang, Jing Xiao

Ping An Technology
{ZHANGCHAO576,CHENGJUNJIE863,GUYANMEI040}@pingan.com.cn

Abstract

In this paper we describe a novel replay detection system for the ASVspoof 2019 challenge. The objective of this challenge is to distinguish arbitrarily audio files from bona fide or spoofing attacks, where spoofing attacking includes replay attacks, text-to-speech and voice conversions. Our replay detection system is a pipeline system with three aspects: feature engineering, DNN models, and score fusion. Firstly, logspec is extracted as input features according to previous research works where spectrum augmentation is applied during training stage to boost performance under limited training data. Secondly, DNN models part includes three major models: SEnet, DenseNet, and our proposed model, channel consistency DenseNeXt, where binary cross entropy loss and center loss are applied as training objectives. Finally, score fusion is applied to all three DNN models in order to obtain primary system results. The experiment results show that for our best single system, channel consistency DenseNeXt, t-DCF and EER are 0.0137 and 0.46% on physical access evaluation set respectively. The performance of primary system obtains 0.00785 and 0.282% in terms of t-DCF and EER respectively. This is a 96.8% improvement compared to the baseline system CQCC-GMM and it achieves state-of-the-art performance in PA challenge.

Index Terms: Replay detection, ASVspoof, automatic speaker verification, spectrum augmentation

1. Introduction

Automatic speaker verification (ASV) system is aiming to verify whether voice is matching to the target speaker. Recently ASV attracts more and more attentions in many real-world applications such as phone unlocking, credit card business, security checking etc. [1, 2], where various ASV technologies have been applied, for example, Gaussian mixture models (GMM) [3], i-vectors [4], x-vectors [5], DNN models [6, 7]. As the number of ASV applications grows, numerous researchers point out the vulnerability of ASV systems to spoofing attacks [8, 9], especially text-to-speech (TTS) [10] and voice conversion (VC) [11] could produce very realistic audios and pass ASV checking, as a result these new technologies bring tremendous challenges to anti-spoof tasks.

There are three ASVspoof challenges until now, the ASVspoof challenge 2015 contains various kinds of TTS and VS attacks, while replay attacks are covered in the ASVspoof challenge 2017 [12, 13]. The ASVspoof challenge 2019 [14] not only includes previous test cases, but also is further promoted, like updating VC and TTS using state-of-the-art methods, applying different quality and distance of replay attacks and re-defining the evaluation metrics. There are two major tasks for ASVspoof challenge 2019, one is known as physical access (PA), where it includes three different quality and three different distance replay attacks, the other is called logical access (LA) which contains more than ten types of different TTS and VS, including the state-of-the-art ones.

For ASVspoof challenge 2019 tasks, most researches can be categorized mainly into two categories. First category is feature engineering [15, 16, 17, 18, 19, 20, 21, 22, 23], including constant Q cepstral coefficients (CQCC), linear frequency cepstral coefficients (LFCCs), Mel-filter frequency cepstral coefficients (MFCCs), inverted Mel-filter frequency cepstral coefficients (IMFCCs), group delay (GD) gram and X-vector embedding as the inputs. Second category is model engineering which includes statistical model and neural networks, for example in [24], adversarial attack is applied in detecting spoofing attacks, [25] explores the performance of squeeze-excitation and residual networks (SEnet), end-to-end DNN model is used in [26], and [27] reveals the anti-spoofing results of raw audio inputs with SincNet and VGG networks.

In this paper, we built a DNN model to further explore anti-spoofing problems based on log power magnitude spectra (logspec) for ASVspoof challenge 2019, our contributions consist of mainly three parts. Firstly, spectrum augmentation [28] and DenseNet network [29] are firstly introduced in this work for ASVspoof challenge. Secondly, we propose a novel model based on modification and combination of DenseNet and ResNeXt [30], named as *channel consistency* DenseNeXt in this work, it reduces both parameters and computing power by half but remains relatively the same performance compared with DenseNet. Lastly, our primary fusion system presents a 96.8% relative improvement compared to CQCC-GMM baseline and outperforms other team in public ranking board in PA challenge according to [14].

The content of this paper is organized as the following. Section 2 demonstrates the details of feature engineering, spectrum augmentation, DNN models that include our proposed channel consistency DenseNeXt as well as score fusion. Section 3 contains brief description of baseline systems and experimental setups. Section 4 presents overall results from baseline system, DNN models and other researchers' works. Finally, the conclusion of this paper is drawn in section 5.

2. Methods

In this section, we first describe feature engineering, where logspec is introduced and used as inputs, and we also briefly introduce MFCC, CQCC and LFCC, where CQCC and LFCC are considered as official baseline. Then we explain details of three different DNN models used in our primary system, including our proposed channel consistency DenseNeXt. Finally we describe spectrum augmentation and score fusion strategy used for the primary system.

http://dx.doi.org/10.21437/Interspeech.2020-1044

2.1. Feature Engineering

Logspec: Kaldi tool-kit [31] is used to extract unified feature maps of logspec based on previous work [32, 25], the utterances are firstly extended to multiple of 400 frames, each segment has a final shape of (400, 257), where 400 belongs to time domain and 257 is frequency domain with 512 fast Fourier transform (FFT) bins, each segment overlaps 200 frames which is same to [32, 25]. As suggested, we do not apply voice activity detection (VAD) or any normalisation. Moreover, the score of one utterance is averaged by all segments for testing and evaluation.
MFCC: Apart from logspec, we use Kaldi tool-kit to extract Mel frequency cepstral coefficients (MFCC) with window length of 25ms and window shifts of 10ms. Hamming window is used and total 23 Mel coefficients are calculated with frequency range lies between 0 to 8000Hz. Normalisation and VAD are not used for MFCC which are same as logspec.
CQCC: The constant Q transform (CQT) was initially proposed for music processing where higher frequency resolution at lower frequency and higher temporal resolution at higher frequency. Constant Q cepstral coefficients (CQCC) provides better performance based on CQT, more details can be found in [15].
LFCC: The detail of linear frequency cepstral coefficients (LFCC) can be found in [33], the extraction process of LFCC is like MFCC but with triangular shape filters.

2.2. DNN models

Remarkable performance for anti-spoofing tasks have been achieved by using DNN models in previous research works [23, 25, 26, 27], so we first introduce SEnet34 [34] as one of our baseline systems, then we introduce DenseNet that is firstly employed to anti-spoofing tasks for ASVspoof 2019 challenge, finally we describe our proposed model, channel consistency DenseNeXt which further improves DenseNet.

2.2.1. SEnet

The most contribution of SEnet with squeeze-and-excitation blocks [34] is that it achieves decent performance boosting in the cost of little increase of parameters. In this work, we have implemented SEnet34 as in [25] where it gives the best result over all other single systems. We train this model with logspec and 64 channels.

2.2.2. DenseNet

We follow the previous work [29] in computer vision and implement DenseNet with three dense blocks. As Figure 1 illustrates, each dense block contains multiple of bottleneck blocks, and each of bottleneck block includes two convolutional layers with kernel size 1 and 3 respectively. Transition layer includes one convolutional layer and one average pooling layer with stride equals two. Layer batch-norm and leaky-ReLU are applied to all convolutional layer. At the end, outputs of two fully connected layers are used for center loss [35] and cross-entropy (CE) loss calculation.

2.2.3. Channel consistency DenseNeXt

Despite the tremendous success of DenseNet in computer vision, however, due to the growing numbers of bottleneck layers and growth of feature maps in DenseNet, parameters and computing power increase rapidly, eventually this brings memory-consuming problem and limits the depth of dense blocks. To

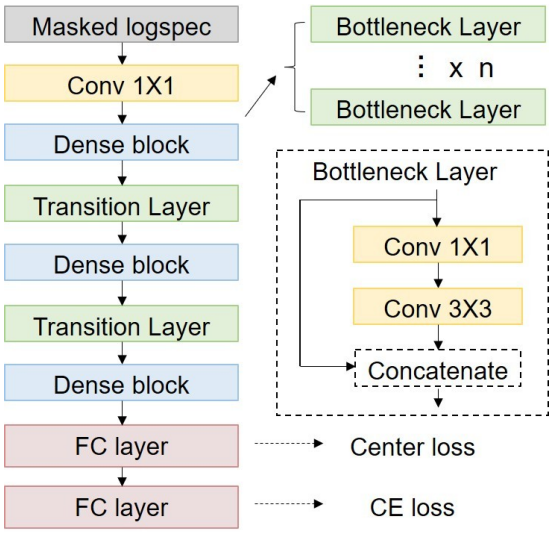

Figure 1: *The structure of DenseNet used in this work, all convolutional layers are followed by batch-norm layer and leaky-ReLU, padding is used if kernel size is not 1. Transition layer contains one convolutional layer and one average pooling layer.*

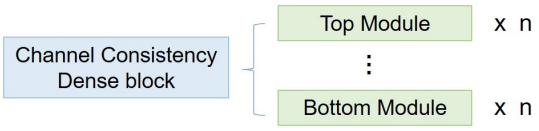

Figure 2: *The difference of DenseNet and DenseNeXt is the dense block part, where the dense block contains multiple bottleneck layers in DenseNet, while in DenseNeXt the Dense block is subsitiuded by channel consistency block consisting of multiple top and bottom modules.*

overcome this problem, we propose a novel structure to replace the original dense blocks. As Figure 2 illustrates, stacked bottleneck layers are replaced by multiple of top and bottom modules, the detail of top and bottom module is presented in Figure 3. Inspired by the work of [30], we place four parallel convolutional layers within the middle of module, the purpose of these parallel layers is to reduce the total computing power. The structure of SE block is identical to [34] where two fully connected layers and sigmoid are applied. Only bottom modules have the first convolutional layer, which is named channel reduction in the Figure 3, the rest parts are identical to both top and bottom modules. Hence instead of using stacked bottleneck layers in DenseNet, the dense block in DenseNeXt contains four top modules in sequence and then following with four bottom modules. The feature maps of input data into every top and bottom modules are increasing, because reduction layer in bottom module reduces twice the number of feature maps than increased in top module, total number of feature maps are decreased in each top module. Using reduction layer can integrate information where network learned and reduce total parameters. Eventually, when the number of top and bottom layers keeps consistent, the numbers of input features maps and output features maps of each dense block will be exactly same. In summary, firstly less feature maps are used in DenseNeXt comparing with

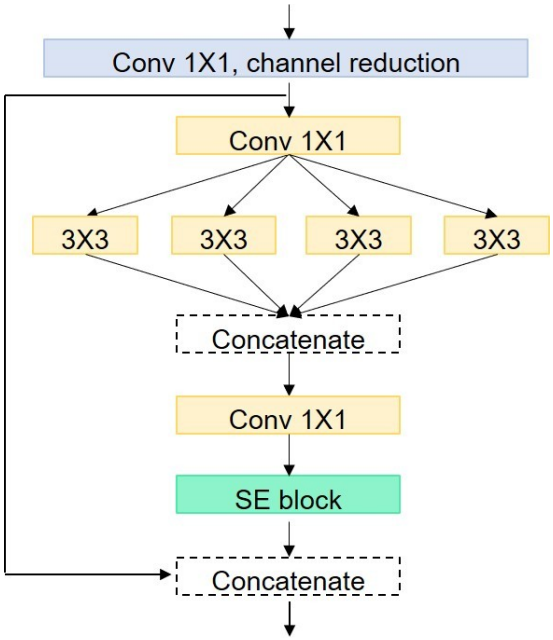

Figure 3: *The structure of bottom modules used in DenseNeXt, in contrast to top modules, the bottom modules contain the special convolutional layer (marked as channel reduction), the rest parts are identical to top and bottom modules.*

DenseNet, secondly applying parallel convolutional layers further reduces total computing power used, finally using SE block will boost performance in cost of few parameters. Consequently this channel consistency DenseNeXt explicitly keeps the similar of network depth within each dense block and reduces both total parameters and computing powers.

2.3. Spectrum Augmentation

Inspired by masking in [36, 28], a certain percentage of logspec features are masked during training stage. In this work, a continuous 30-dimensions out of 257 feature dimensions are randomly selected and set to 0 for each mini-batch. We believe that by doing so, it can increase the model generalisation performance with limited training data.

2.4. Score Fusion

For the primary system, we average the scores over different models as we find that each model could overfit some part of training dataset, score averaging strategy is very beneficial to final performance. The primary system contains the scores from three models: SEnet34, DenseNet and channel consistency DenseNeXt. Because the performance of SEnet34 is worse than the other two models, we first half SEnet34 scores and then average all three models.

3. Experimental Setups

3.1. Baseline System

The official baseline system of CQCC-GMM and LFCC-GMM are extracted followed by official script. Both features include static, delta and delta delta coefficients [15]. GMM [3] contains two mixtures with 512 dimensions of each mixture.

3.2. Experimental Setup

3.2.1. Dataset and Evaluation metrics

Physical Access: There are 5400 bona fide audios and nine types of different distance and quality replay attacks, where there are 54000 audios in total for training. There are total 29700 and 134730 audios in development and evaluation set respectively.

Logical Access: Training set contains 2580 bona fide audios and 22800 faked audios made from six different TTS and VS technologies, where same technologies are used in development set with 24844 audios in total, including bona fide ones. However extra 13 unknown types of spoofing are involved in evaluation set with total 71273 audios.

Evaluation metrics: Our goal is to minimise normalized tandem decision cost function (t-DCF) and equal error rate (EER) defined by organizer [37], an official evaluation script is provided by organizer and it is used to verify performance for all results.

3.2.2. Network Optimization

For each dense block in DenseNet, 11 layers of bottleneck layers, or total 22 convolutional layers are used. The numbers of top and bottom modules used in channel consistency DenseNeXt are both 4 in each block, so there are 24 convolutional layers in total. Initial feature maps are set to 8 and the growth rate is set to 4 for both networks, the reduction rate of SE block is set to 4 instead of 16 in DenseNeXt because we set initial feature maps to 8. The configuration of SEnet34 is the same as in [34]. Adam optimizer is applied with β_1 = 0.9 and β_2 = 0.98 for all three models. Meanwhile 1e-4 of weight decay rate is applied to reduce the overfitting problem. Moreover, the learning rate is initially set to 5e-4 and lately adjusted according to accuracy of testing set, to be specific, we used ReduceLROnPlateau API in Pytorch to adjust learning rate where metric is testing accuracy, mode is set to 'max' and patience is set to 5, therefore learning rate will be reduced by factor 2-10 as soon as testing set accuracy is not reaching new maximum after 5 epochs.

The output dimension of last second fully-connected layer is set to 128 and it is used in center loss calculation according to [35], the output of last fully-connected layer is 2 which stands for bona fide or spoofing. The total loss is a combination of two losses where weight ratio between CE loss and center loss is 20:1. We train all DNN models for 100 epoch and keep one model with the lowest t-DCF tested on development set. It is worth mentioning that we used stride step of one in first convolutional layer rather than two in DenseNet and our proposed model, this is because we find that it is harmful to final performance with early reduction of input dimensions through experiments, although this increases computing power significantly. Fortunately we manage to reduce total computing power through other ways.

4. Results and Discussion

In this section we evaluate both baseline system CQCC-GMM and LFCC-GMM with script prepared from the organizer. The results of logspec over SEnet34, DenseNet and DenseNeXt are also measured and presented. In order to show the difference between our proposed model and other models, we calculate the parameters and computing power of each model. Moreover, we test different features including CQCC, MFCC and logspec

Table 1: *Results on unified feature maps of logspec over different models. The word 'masked' denotes that spectrum augmentation is applied during training stage. Other models are trained without spectrum augmentation in default.*

System	PA development		PA evaluation	
	t-DCF	EER	t-DCF	EER
CQCC-GMM	0.195	9.87	0.245	11.04
LFCC-GMM	0.255	11.96	0.302	13.54
SEnet34(A)	0.0305	1.185	0.0509	1.929
DenseNet	0.0146	0.556	0.0243	0.890
masked DenseNet(B)	0.0118	0.447	0.0182	0.657
DenseNeXt	0.0116	0.462	0.0197	0.736
masked DenseNeXt(C)	**0.0082**	**0.314**	**0.0137**	**0.465**
Fusion(B+C)	0.0051	0.185	0.0091	0.310
Fusion(A+B+C)	**0.0042**	**0.152**	**0.00785**	**0.282**

Table 2: *Results of performance on three models in terms of parameters and computing power. The units of parameters and flops are kilo and giga.*

System	Parameters(k)	flops(G)
SEnet34	22635	7.82
DenseNet	171	7.16
DenseNeXt	82	3.53

on the same model for the ASVspoof tasks. Finally results of logical access over three DNN models are presented in the end.

4.1. Results on Logspec

Table 1 shows the results on baseline systems and DNN models based on logspec, the fusion system is a combination of SEnet34, DenseNet and DenseNeXt where the score of SEnet34 is halved. Our proposed single system (DenseNeXt) shows the best performance over other systems and gains a relative 94.4% improvement compared to baseline system CQCC-GMM on evaluation set in terms of t-DCF. Meanwhile, we compare the impact of applying spectrum augmentation during training stage. It can be seen that spectrum augmentation increases the generalization and produces a less overfiting model where t-DCF is further reduced by 28%. Table 2 shows the total parameter numbers and computing power of each model. It is clear that our proposed model achieves 50% reduction of parameters and computing power while yields better result than DenseNet.

4.2. Results on Different Features

As we have shown the best single system in this work is channel consistency DenseNeXt according to Table 1, in order to verify the effect of using different features, we choose three different features trained on the same model to compare the performance, the results are presented in Table 3. Obviously, logspec outperforms other features with giant performance gap for ASVspoof tasks.

Table 3: *Results of different features on DenseNeXt model in terms of t-DCF and EER, no spectrum augmentation applied.*

DenseNeXt	PA development		PA evaluation	
	t-DCF	EER	t-DCF	EER
CQCC	0.1741	7.486	0.2480	10.210
MFCC	0.1222	4.353	0.1393	4.976
logspec	**0.0116**	**0.462**	**0.0197**	**0.736**

Table 4: *The ASVspoof 2019 PA scenario evaluation results of our primary system and other teams [14]*

ASVspoof 2019 PA scenario		
ID	t-DCF	EER
Ours	**0.00785**	**0.28**
T28	0.0096	0.39
T45	0.0122	0.54
T44	0.0161	0.59
T10	0.0168	0.66
T24	0.0215	0.77
T53	0.0219	0.88
T17	0.0266	0.96

Table 5: *Results for logical access. Unified feature maps of logspec are extracted for DNN models, the score of a whole utterance is the average score for all belonging segments.*

System	LA development		LA evaluation	
	t-DCF	EER	t-DCF	EER
CQCC-GMM	0.066	2.71	0.212	8.09
SEnet34(A)	0.000	0.00	0.202	9.94
DenseNet(B)	**0.000**	**0.00**	**0.187**	**6.78**
DenseNeXt(C)	0.000	0.04	0.202	8.63
Fusion(A+B+C)	**0.000**	**0.00**	**0.119**	**4.46**

4.3. Results Comparing with Previous works

Table 4 shows the results of our primary system and other research works give by [14], apparently our primary system outperforms other models that achieves the new state-of-the-art result, where it further decreases the t-DCF by relative 18% when compared with team T28. Furthermore, the performance of our best single system shows the potential over other fusion systems. Table 5 shows results for logical access, the t-DCF of primary system beats the baseline while single system gets the slightly better performance compared to baseline systems.

5. Conclusions

In this work we further explore the potential of DNN models for anti-spoofing tasks. We evaluate three DNN models, SEnet34, DenseNet as well as channel consistency DenseNeXt, over unified feature maps of logspec. The experimental results show that spectrum augmentation is beneficial and further reduces t-DCF. We compare our proposed model channel consistency DenseNeXt with DenseNet in terms of parameters and computing power, it is shown that our proposed model is able to gain performance improvement with less parameters and computing power. We also investigate different features trained on the same model, including MFCC, CQCC and logspec. Experimental results show the robustness of logspec over other features for anti-spoofing tasks. Our single and primary systems dramatically improve the result of t-DCF and EER over both development and evaluation of PA dataset for ASVspoof 2019 challenge.

6. References

[1] K. A. Lee, A. Larcher, H. Thai, B. Ma, and H. Li, "Joint application of speech and speaker recognition for automation and security in smart home," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and

humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] 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.

[4] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 225–229.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[7] P. Kenny, T. Stafylakis, P. Ouellet, V. Gupta, and M. J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition." in *Odyssey*, vol. 2014, 2014, pp. 293–298.

[8] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015, pp. 1–6.

[9] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning Wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[11] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," *Interspeech*, 2017.

[12] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. W. Evans, J. Yamagishi, and K.-A. Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection." in *Interspeech*, 2017, pp. 2–6.

[14] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," *Interspeech*, 2019.

[15] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients." in *Odyssey*, vol. 45, 2016, pp. 283–290.

[16] M. J. Alam, G. Bhattacharya, and P. Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization." in *Odyssey*, 2018, pp. 393–398.

[17] G. Suthokumar, V. Sethu, C. Wijenayake, and E. Ambikairajah, "Modulation dynamic features for the detection of replay attacks." in *Interspeech*, 2018, pp. 691–695.

[18] D. Li, L. Wang, J. Dang, M. Liu, Z. Oo, S. Nakagawa, H. Guan, and X. Li, "Multiple phase information combination for replay attacks detection." in *Interspeech*, 2018, pp. 656–660.

[19] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[20] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7234–7238.

[21] T. Gunendradasan, B. Wickramasinghe, P. N. Le, E. Ambikairajah, and J. Epps, "Detection of replay-spoofing attacks using frequency modulation features." in *Interspeech*, 2018, pp. 636–640.

[22] J. Williams and J. Rownicka, "Speech replay detection with x-Vector attack embeddings and spectral features," *Interspeech*, 2019.

[23] W. Cai, H. Wu, D. Cai, and M. Li, "The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion," *Interspeech*, 2019.

[24] S. Liu, H. Wu, H.-y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[25] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-spoofing with squeeze-excitation and residual networks," *Interspeech*, 2019.

[26] J.-w. Jung, H.-j. Shim, H.-S. Heo, and H.-J. Yu, "Replay attack detection with complementary high-resolution information using end-to-end dnn for the asvspoof 2019 challenge," *Interspeech*, 2019.

[27] H. Zeinali, T. Stafylakis, G. Athanasopoulou, J. Rohdin, I. Gkinis, L. Burget, J. Černockỳ *et al.*, "Detecting spoofing attacks using VGG and SincNet: But-omilia submission to asvspoof 2019 challenge," *Interspeech*, 2019.

[28] 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," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[30] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[32] C.-I. Lai, A. Abad, K. Richmond, J. Yamagishi, N. Dehak, and S. King, "Attentive filtering networks for audio replay attack detection," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6316–6320.

[33] M. Sahidullah, T. Kinnunen, and C. Hanilci, "A comparison of features for synthetic speech detection," *Interspeech*, 2015.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[35] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Cision*, 2016, pp. 499–515.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171–4186, 2019.

[37] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," *Odyssey*, 2018.