2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

ResMax: Detecting Voice Spoofing Attacks with Residual Network and Max Feature Map

Il-Youp Kwak
Chung-Ang University
Seoul, Korea
ikwak2@cau.ac.kr

Sungsu Kwag
Samsung Research
Seoul, Korea
sungsu.kwag@samsung.com

Junhee Lee
Samsung Research
Seoul, Korea
jh0913.lee@samsung.com

Jun Ho Huh
Samsung Research
Seoul, Korea
junho.huh@samsung.com

Choong-Hoon Lee
Samsung Research
Seoul, Korea
choonghoon.lee@samsung.com

Youngbae Jeon
Korea University
Seoul, Korea
jyb9443@korea.ac.kr

Jeonghwan Hwang
Korea University
Seoul, Korea
ju.su.splab@gmail.com

Ji Won Yoon
Korea University
Seoul, Korea
jiwon_yoon@korea.ac.kr

Abstract—The “2019 Automatic Speaker Verification Spoofing And Countermeasures Challenge” (ASVspoof) competition aimed to facilitate the design of highly accurate voice spoofing attack detection systems. the competition did not emphasize model complexity and latency requirements; such constraints are strict and integral in real-world deployment. Hence, most of the top performing solutions from the competition all used an ensemble approach, and combined multiple complex deep learning models to maximize detection accuracy – this kind of approach would sit uneasily with real-world deployment constraints. To design a lightweight system, we combined the notions of skip connection (from ResNet) and max feature map (from Light CNN), and evaluated the accuracy of the system using the ASVspoof 2019 dataset. With an optimized constant Q transform (CQT) feature, our single model achieved a replay attack detection equal error rate (EER) of 0.37% on the evaluation set, surpassing the top ensemble system from the competition that achieved an EER of 0.39%.

Index Terms—voice assistant security, voice spoofing attack, voice synthesis attack, voice presentation attack detection

I. Introduction

Voice assistants are quickly being upgraded to support security- and privacy-critical commands such as sending and receiving emails, taking photos and posting on social media, and making payments. Provision for such advanced features make voice assistants lucrative targets for attackers to exploit. “Voice spoofing attacks” (also referred to as voice replay attacks) are perhaps the easiest and most accessible way for attackers to exploit voice assistants. These attacks involve recording a victim’s use of voice assistants, and simply replaying them through a loudspeaker to bypass voice biometric authentication services enabled on the victim’s target devices. “Voice synthesis (or conversion) attacks” are more advanced attacks, which involve collecting the victim’s voice samples, training the victim’s voice biometric models using machine learning techniques, and generating new voice attack samples. There are free tools available for training voice models, including Google’s Wavenet or Tacotron [1], [2]. “Automatic speaker verification spoofing and countermeasures challenge” (ASVspoof) competitions have been organized to encourage researchers to develop voice spoofing attack detection systems and to compete for the improvement of voice detection accuracy. Competitions were held in 2015, 2017, and 2019 [3]–[5] – new training sets and evaluation sets have been released for each competition.

In this study, we focus on the latest competition, ASVspoof 2019 dataset, and compare the performance of our solution against the best performing solutions from that competition. ASVspoof 2019 provided two different types of attack sets: (1) physical access (PA) attack set that is generated from replaying recorded voices through loudspeakers, which designed to prevent “Voice spoofing attacks”, and (2) logical access (LA) attack set that involves synthetic generation by training victims’ voice models, and playing the trained voice models directly to the target voice assistant system, which designed to prevent “Voice synthesis attacks”; thus, there is no recording nor replaying involved in the generation of the second set. The best performing model against the ASVspoof 2019 PA set achieved an equal error rate (EER) of 0.39%. An EER represents an error rate at which the rate of mis-classifying genuine live samples as attacks is equal to the rate of mis-classifying attack samples as genuine live samples. The participants, however, used an ensemble approach comprising of multiple deep learning models [5] Another ensemble solution that combined multiple light CNN (LCNN) models achieved an EER of 0.54% on the PA set, and and EER of 1.86% on the LA set [6]. An ensemble solution that used multiple residual network (ResNet) models with the squeeze and excitation (SE) technique achieved an EER of 0.59% for the PA set, and 6.7% for the LA set [7].

Despite their competitive accuracy results (low EERs), such ensemble (multi-model) solutions can be challenging with latency and model complexity requirements imposed by real life businesses. Owing to the users’ timely response expectations and exploding server cost issues, businesses typically require model sizes to be less than a few megabytes (also considering on-device deployment scenarios), and detection (prediction) time to be less than 100 ms.

To satisfy those business requirements, we propose a deep learning architecture called "ResMax" that combines the skip connection notion from ResNet and with the max feature map notion from LCNN. By using our optimized constant Q transform (CQT) feature, ResMax achieved an EER of 0.37% on the PA set and 2.19% on the LA set. Compared to the top performing solutions from the ASVspoof 2019 competition that used ensemble approaches (multiple heavy and complex deep learning models), our ResMax single model is capable of achieving the top EER on the PA set, and would be ranked third among the LA set solutions. For comparison, the top performing single model solution that used an LCNN architecture with fast Fourier transform as a feature achieved an EER of 4.53% on the LA set. As for the PA set, the top performing single solution achieved approximately 0.5% EER [5].

II. Methodology

A. Feature Engineering

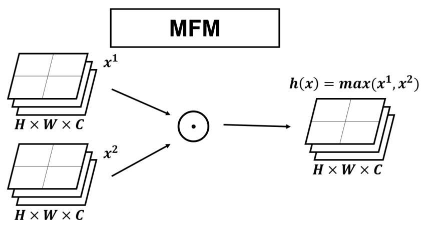
We initially experimented with two most used spectral features in the ASVspoof 2017 and 2019 competitions: (1) constant Q transform (CQT), and (2) short time Fourier transform (STFT). [4], [5], [8] After a few trials and observing model accuracy, we noticed that CQT was more compatible with the proposed model, demonstrating significantly superior model accuracy. Hence, we focused on further tuning CQT parameters to optimize model accuracy. CQT explores an audio signal $x[n]$ in time-frequency representation by dividing the signal into shorter frames, and analyzing the audio in frequency domain. The CQT output in discrete form is as follows:

$$X_k = \sum_{n=0}^{N_k-1} W_{k,n} x_n e^{-j2\pi Qn/N_k}, \tag{1}$$

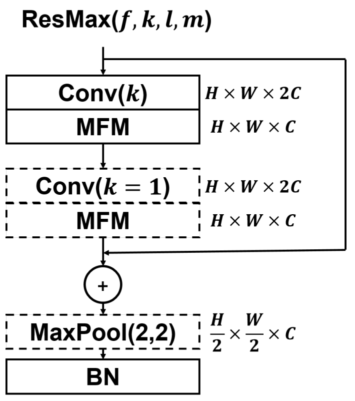
where k is the index of a frequency bin, which ranges from 1 to the total number of frequency bins (K); N_k is the frame size of kth bin, and W is the windowing function that is used for tapering each frame; and Q is the quality factor that determines the feature resolution. To transform a given data series into a frequency domain, CQT applies filters as center frequency f_k and bandwidth of B_k in kth frequency domain. The center frequency of the kth filter is $f_k = (2^{1/C})^k f_{min}$, where f_{min} is the bandwidth of the lowest frequency, and C is the number of octaves in each filter. Q is determined as f_k/B_k.

For CQT optimization, we attempted frequency variations in f_{min} and K parameters. The highest center frequency, f_K, was determined by $(2^{1/C})^K f_{min}$. We tested two f_{min} values, 1Hz and 32Hz, to analyze the effects of altering low frequency components. We also experimented with variations of K to try f_K values at 323Hz and 1024Hz to see the effects of high frequency components. For fair comparison with constant resolution in the time and frequency domain, we fixed the number of bins per octave and hop size to 12 and 512, respectively. We used Hann window as the windowing function.

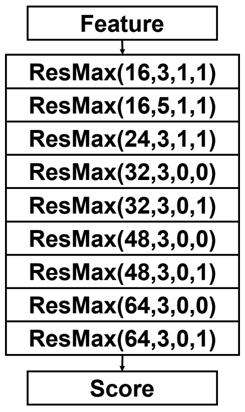
As for the sample lengths (duration) we used 9 seconds. For samples that were longer than 9 seconds, we used the first 9 seconds of that sample. For samples that were shorter than 9 seconds, we replicated the samples and repeatedly concatenated them until we had a 9-second sample. We did not employ any normalization method.



(a) MFM



(b) ResMax Block



(c) Model Architecture

Fig. 1. ResMax architecture descriptions: (a) represents an MFM layer; (b) represents a ResMax block; (c) describes the building block for the entire model architecture. A ResMax block has four parameters: f is the number of ResMax filters, k is the kernel size (k, k) in convolution layer. l is 1 if a ResMax block has an additional convolution layer with a following MFM layer (dotted Conv and MFM blocks in (b)), and 0 otherwise. m is 1 if a ResMax block has an additional max pooling layer (dotted MaxPool block in(b)), and 0 otherwise.

B. ResMax: Residual Network with Max Feature Map

Both LCNN and ResNet based models have performed well against the ASVspoof 2017 and ASVspoof 2019 evaluation datasets, demonstrating top performances with respect to EERs [6], [7], [9], [10]. However, all of the top performing solutions to date have used the ensemble approach, combining multiple deep learning models to minimize error rates. Such models do not consider model latency and complexity requirements as explained in Section I. To design a lighter model, we propose "Residual network with Max feature map" (ResMax) blocks that integrate the max feature map (MFM) notions from LCNN and skip connection notions from ResNet. MFM layer is used after a convolution layer (see Fig. 1 (a)). MFM initializes two layers with the same dimensions, and selects the maximum of two items from the same position in the two layers. This process improves the robustness of the model, and makes the model lighter through filter selection. [11]

A ResMax block that uses the skip connection network is described in Fig. 1 (b). A skip connection adds original input to a processed network, $F(x)$, where output = input + $F(input)$ (See Fig. 1 (b)). Here, the training weight parameters in $F(input)$ imply training the residual in output − input. This residual part would often have values close to zero, and helps with solving the vanishing gradient problem during back propagation. Further, the 2 by 2 MaxPooling technique used in a ResMax block reduces the number of parameters and the model complexity as a result.

The full model architecture comprising of 9 ResMax blocks is shown in Fig. 1 (c). A flatten layer follows the last ResMax block. We then have a dropout layer, and a dense layer that outputs the results using the softmax function.

To demonstrate the performance of the ResMax model, it is necessary to compare it with existing models that performed well in the ASVspoof 2019 challenge.

Table I describes the top performing ensemble models that participated in the ASVspoof 2019 [6], [12]–[15]. The T10 model ranked fourth for the PA data with ResNet Architecture, which has 36 layers with the data augmentation technique. The T10 model was an ensemble model of 6 ResNet models with different features such as LFCC, IMFCC, STFT, GD gram, and Joint gram [13]. The T44 model ranked third in the PA data using CQCC and logspectogram features with SENet, ResNet, and Dilated ResNet architectures. Five models were ensembled in the T44 model, and the single best model used logspectogram feature with SENet34 [12]. The T45 model ranked second in both the LA and PA data using an ensemble approach with features such as LFCC, CQT and FFT. Their FFT-LCNN model was the best performing single model for the LA data, and CQT-LCNN was the best performing single model for the PA data [6]. The T50 model ranked 5th in the LA data, and attempted data augmentation using variational autoencoder on CQT feature [14]. The T60 model ranked third in the LA data. It used FFT feature with CNN, CRNN, 1D-CNN, and Wave-U-Net and shallow models of IMFCC-GMM and ivector-SVM [15].

We used the development and evaluation set of ASVspoof2019 competition data to compare the performance of ResMax model with the ensemble models described in Table I.

TABLE I
Ensemble solutions from ASVspoof 2019 and the list of models used.

Model	Data	All models used
T10 [13]	PA	LFCC-ResNet, GD gram-ResNet .. Joint gram-ResNet
T44 [12]	PA	logspec-SENet34, CQCC-ResNet .. logspec-SENet50
T45 [6]	LA	LFCC-LCNN, LFCC-CMVN-LCNN .. CQT-LCNN
T50 [14]	PA	CQT-LCNN, LFCC-LCNN, DCT-LCNN
	LA	CQT-CGCNN, CQT-ResNet18 .. CQT-ResNet18IVec
T60 [15]	PA	FFT-CNN, FFT-CRNN, IMFCC-GMM, SVMs-IVec

C. Hyper Parameter Tuning

To handle unequal distribution of attack samples and live genuine samples in the ASVspoof 2019 train set (i.e., an imbalanced dataset issue), we applied cost-sensitive learning method [16]. Our primary evaluation criterion was EER that considered attack samples and genuine samples with same importance. Thus, we multiplied the minority class, genuine samples, with weight parameters, ensuring that the minority class also equally influenced the classification model. The overall training latency did not change as it merely multiplies parameters in the loss function. We used the grid search method to select an optimal parameter set that would minimize the EER on the development set. The optimized set of hyper parameters includes feature extraction parameters, cost-sensitive learning parameters, classification algorithm specific parameters, and regularization parameters.

III. Experiments

A. Experimental Setup

For evaluation, we used the LA and PA data from the ASVspoof 2019 competition [5]. Each data consists of a training set, a development set, and an evaluation set. We trained ResMax models using the training sets, and evaluated the trained models on the development set and evaluation set.

To measure the detection accuracy, we primarily used EER, following the rules from the ASVspoof 2019 competition. The mis-classifying error rate for the loudspeaker was referred to as "false rejection rate" (FRR), and the mis-classifying error rate for the human voice was referred to as the "false acceptance rate" (FAR). Our binary classifiers return a probability score that represents the likelihood of a command being on a loudspeaker, computing the probability scores for every command in a given evaluation set. There is always a trade-off between FAR and FRR, and adjusting the threshold value would always lower one error rate at the cost of increasing the other error rate. A value at which the FAR is equal to the FRR for a given command set determines the EER for that given model.

In addition, we used the minimum normalized tandem detection cost function (t-DCF) [17], which was also used (and reported) in the competition. A fixed automatic speaker verification system provided from the competition was used to calculate t-DCF values.

We ran 100 epochs to train the proposed models, and performed 10 training and evaluation sessions – this is because the models would converge to slightly different parameters for each run.

B. Hyperparameter Selection

This section describes how we selected the main hyperparameters of the ResMax model. We fitted our model 10 times with the proposed hyperparameters and evaluated averaged EER on the development and evaluation data set for each of the LA and PA data.

1) CQT Parameter Selection: We experimented with three different ResMax models based on two different CQT parameters, f_{min} and K: CQT-1_100-ResMax, CQT-1_120-ResMax, and CQT-32_60-ResMax. The first number in the model name represents the f_{min} value, and the second number represents the K value.

Fig. 2 shows the average EERs for CQT-1_100-ResMax, CQT-1_120-ResMax, and CQT-32_60-ResMax on development and evaluation set for the LA and PA data. CQT-1_100-ResMax showed the best performance for the LA data, and CQT-1_120-ResMax showed the best performance for the PA data. CQT-1_120-ResMax, CQT-1_100-ResMax, and CQT-32_60-ResMax showed statistically significant differences in LA data for both the development and the evaluation sets ($p < 0.01$ two sample t-tests with Bonferroni correction). For PA data, both CQT-1_120-ResMax and CQT-1_100-ResMax have a better statistical significance than CQT-32_60-ResNet model in the development and evaluation sets ($p < 0.01$ two sample t-test with Bonferroni correction). However, no statistical significance was detected between CQT-1_120-ResMax and CQT-1_100-ResMax. CQT-1_100-ResMax includes low center frequency(1-32Hz) components and CQT-32_60-ResNet includes high frequency (323 Hz–1024 Hz) components, whereas CQT-1_120-ResMax model includes both high and low frequency. While both high and low frequency components are important, we can infer that the low frequency components are more effective in both PA and LA.

2) Non-Speech Part Removal: There are silent parts at the beginning and at the end of any given speech sample. We designed a simple non-speech part remover based on sound loudness. When our remove is applied, a voice sample as shown in (a) of Fig. 3 would be processed to become (b). Our intuition is that since the non-speech parts would not have much information, the model performance may improve if we only use the speech part to train and test models. Thus, we examined two ResMax models – one trained with the original samples,

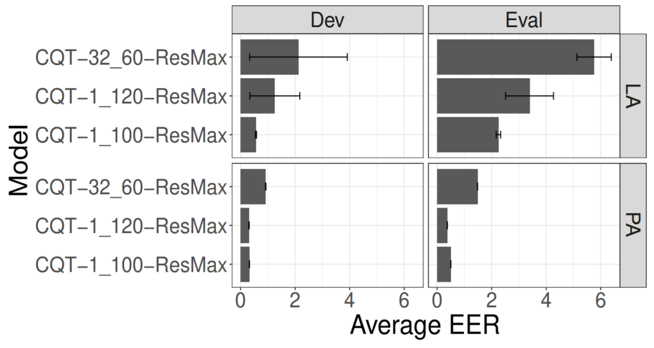


Fig. 2. The CQT-1_100-ResMaX performed best in LA data and the CQT-1_120-ResMax performed best in PA data. The barplot indicates the averaged EER with one standard deviation error bar.

and another trained with the silence-removed samples – for each of the LA and PA data.

Fig. 3 (c) shows the average EER of best performing ResMax models in LA and PA data with and without silence removal. As a result, we found that it is better to train the model with full sound, comprising of both speech and non-speech parts (all $p < 0.01$ with two sample t-tests with Bonferroni correction). These results may indicate that the ResMax model trained to detect even the non-speech part in an electronic speaker's ultra sound.

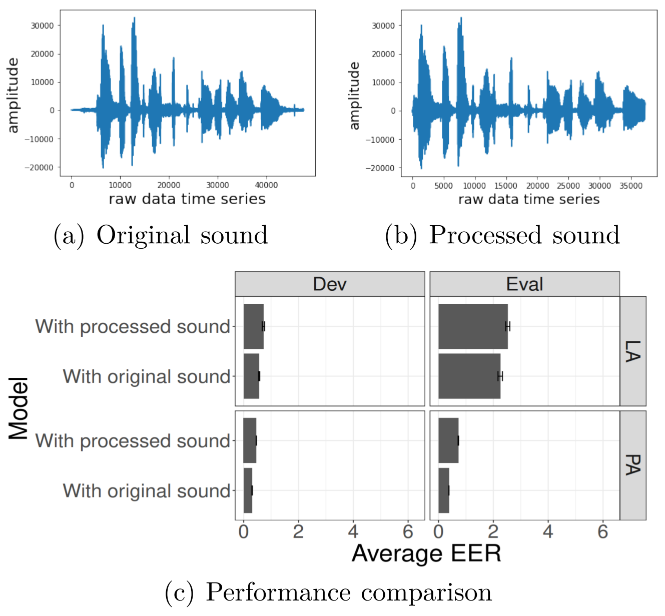


Fig. 3. The non-speech part remover suggested and tested. The ResMax model worked better without the non-speech part remover. The barplot indicates averaged EER with one standard deviation error bar.

3) Sample Duration Selection: Another hyperparameter one could consider optimizing is the sample length: in general, we imagine that the overall performance would improve with increasing sample input time. However, when considering the real-world use of voice

assistants that accept both short commands as well as long commands, it is important for models to achieve high accuracy regardless of the input time. To test our model's performance against both short and long samples, we experimented with different input times, varying between of 3 seconds, 6 seconds, and 9 seconds.

Fig. 4 shows the average EER for CQT-ResMax models trained with 3-second, 6-second, and 9-second samples. We used the best performing CQT-ResMax model in the LA and PA data. The 9-second model showed the best performance for both the development and evaluation sets in the LA and PA data. All pairwise difference in the PA set for both development and evaluation sets were statistically significant (all $p < 0.01$ t-test with Bonferroni correction). As for the LA set, only the 3-second model and the 9-second model showed statistically significant difference ($p < 0.01$ t-test with Bonferroni correction).

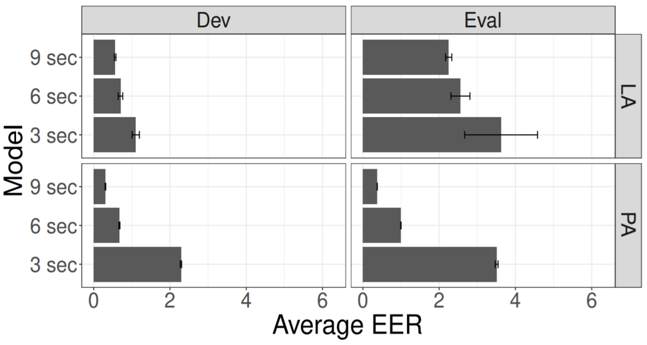


Fig. 4. The 9-second model performed best for both development and evaluation sets in LA and PA data. The barplot indicates averaged EER with one standard deviation error bar.

4) Other Hyper-parameters: Since the number of attack samples is more than 10 times larger than the number of genuine samples in both the LA and PA data, we used cost-sensitive learning by multiplying the minority class (genuine command) with certain degrees of weights in the loss function, ensuring that the minority class also equally influences the classification model. We set this weight as 3. We used binary cross entropy error loss with an Adam optimizer [18]. Initial weights were set using Glorot's uniform initializer [19]. The initial learning rate was 10e-1; we decreased the learning rate to 10e-5 by using learning rate scheduler with sigmoidal decay function. The dropout rate was set to 0.7.

C. Experimental Results

In Table II, we rank our ResMax models against the top five performers from ASVspoof 2019 (including both multi-model and single-model solutions) with respect to evaluation set EERs. We also present the EERs of the top performing single-model solutions in the last few rows – these do not have rankings, and are in italics. As the results indicate, CQT-ResMax outperformed all other single models with respect to both EERs and t-DCFs for both LA and PA evaluation sets. Its EER superiority against the best performing PA and LA single models were all statistically significant ($p < 0.0001$, one sample t-test). The evaluation performed on the development set also showed that the CQT-ResMax model performs the best for both the LA and PA data with respect to EERs; however, the T45's single model performed better with respect to the t-DCF score on the LA data. As for the PA data, our CQT-ResMax model outperformed all other single models.

CQT-1_120-ResMax outperformed all ensemble solutions with respect to both EERs and t-DCFS for the PA evaluation set, achieving an EER of 0.37%. Its EER superiority against T28 (best performing ensemble solution) was statistically significant ($p < 0.0001$, one sample t-test). The CQT-1_120-ResMax model ranked fifth among all ensemble models in the development set with an EER of 0.31%. All other models had a lower EER in the development set compared to the ResMax model – showing higher EERs in the evaluation set indicate that other models would have been overfitted to the development set. We carefully surmise that the CQT-ResMax model is more robust against overfitting issues compared to other ensemble models.

As for the LA set, CQT-1_100-ResMax ranked third with an EER of 2.19% on evaluation set, and ranked fourth with an EER of 0.56% on the development set among all ensemble solutions . It did not show statistically significant superiority against the 4th ensemble solution T60 with respect to EERs but showed statistically significant difference against the 5th ensemble solution T24 in evaluation set ($p < 0.0001$, one sample t-test).

As for ensemble solutions, we simply added the number of parameters used across all models; we present the parameter numbers only for the numbers available in published papers. Considering that CQT-ResMax is a single model solution with far less number of parameters (representing model complexity and computational latency) compared to ensemble solutions, both the LA and PA accuracy results are significant achievements and competitively high.

D. Incorrectly Classified Samples

To explore misclassified samples for CQT-1_120-ResMax in the PA data, we observed the performance of the development set based on environmental conditions. Table III shows how EER performance changes depending on the differences in the verification and the recording environments of the development set. We trained 10 CQT-1_120-ResMax models, and recorded the average EER values for each environmental condition. The most affected factor was replay device quality. When the quality of the replay device was A (perfect), the performance was the worst with an averaged EER of 0.0067, and the performance improved with declining quality of the replay device. The more attackers perform replay attacks with

TABLE II
ResMax performance on the ASVspoof 2019 development sets and evaluation sets. The EERs and t-DCF results are compared against the top five models from each dataset; models are sorted based on the EER in a descending order. Systems that use a single model are in italic. #Mo describes the number of models used in an ensemble system. Two top performing single model systems are also shown at the end of each table. #Params represents the number of parameters contained in the models. (Results that are not public are denoted as hyphens)

LA

#	Model	t-DCF (Dev)	EER (Dev)	t-DCF (Eval)	EER (Eval)	#Mo	# Params
1	T05	-	-	0.0069	0.22	-	-
2	T45	0.0000	0.000	0.0510	1.86	5	1484K
3	CQT-1_100-ResMax	0.0179	0.56	0.0600	2.19	1	262K
4	T60	0.0	0.0	0.0755	2.64	4	-
5	T24	-	-	0.0953	3.45	-	-
6	T50	0.027	0.90	0.1118	3.56	-	-
	T45 (FFT-LCNN)	0.0009	0.040	0.1028	4.53	1	371K
	T45 (LFCC-LCNN)	0.0043	0.157	0.1000	5.06	1	371K

PA

#	Model	t-DCF (Dev)	EER (Dev)	t-DCF (Eval)	EER (Eval)	#Mo	# Params
1	CQT-1_120-ResMax	0.0066	0.31	0.0091	0.37	1	262K
2	T28	-	-	0.0096	0.39	-	-
3	T45	0.0001	0.0154	0.0122	0.54	3	1113K
4	T44	0.003	0.129	0.0161	0.59	5	5811K
5	T10	0.0064	0.24	0.0168	0.66	6	1330K
6	T24	-	-	0.0215	0.77	-	-
	T28	-	-	-	0.50	1	-
	T45 (CQT-LCNN)	0.0197	0.800	0.0295	1.23	1	371K
	T44 (logspec-SENet)	0.015	0.575	0.0360	1.29	1	1344K

high-quality speakers, the harder it would be to detect. The reverberation time (T60) and room size factors that cause the channel effect did not have a significant impact on performance. Also, when the T60 reverberation time was small, in the ranges of 50–200 ms (A), the EER was relatively higher at 0.0017. Another interesting result is that the model have slightly higher risk when the talker-to-ASV distance or attacker-to-talker distance is far enough. When the talker to ASV distance or attacker-to-talker distance is near (A, 10–50 cm) or far (C, >100cm), the EER was higher than medium distance (B, 50-100 cm).

TABLE III
Detection performance on the ASVspoof2019 Physical Access evaluation sets in various environments. The A, B, C represent the classes of each factor which is well described in [5]. All numerical values represent the average of EER.

	Factors	A	B	C
Verification Env.	Room size (S)	0.0047	0.0044	0.0041
	T60 (R)	0.0055	0.0029	0.0038
	Talker-to-ASV distance	0.0059	0.0036	0.0042
Recording Env.	Attacker-to-talker distance (D_a)	0.0051	0.0036	0.0041
	Replay Device Quality (Q)	0.0067	0.0036	0.0009

In the LA data, there are a total of 17 attack types, 6 of which are used in the training and development sets, and are considered as the known attack set. Two known attacks (A16, A19) and 11 unknown attacks are given to the evaluation set. Table IV and Fig. 5 show the averaged EER values for 13 attack types in the evaluation set. The A08, A17, A18, and A19 attack types have high EER. Among them, A17, A18 and A19 were attacks using only voice conversion. According to [5], the variation of averaged EER is bigger than the variations in the other sets. Although the audio samples from A19 are already trained in the training set, it shows relatively higher EER than other attacks.

TABLE IV
Detection performance on the ASVspoof2019 Logical Access evaluation sets in various attacks. Detailed description of each attack is in [5]. All numerical values represent the average of EER.

ID	Type	Description	EER
A07	TTS	vocoder + GAN	0.0022
A08	TTS	neural waveform	0.0388
A09	TTS	vocoder	0.0003
A10	TTS	neural waveform	0.0045
A11	TTS	griffin lim	0.0039
A12	TTS	neural waveform	0.0002
A13	TTS,VC	waveform concatenation & filtering	0.0051
A14	TTS,VC	vocoder	0.0012
A15	TTS,VC	neural waveform	0.0030
A16	TTS	waveform concatenation	0.0039
A17	VC	waveform filtering	0.0561
A18	VC	vocoder	0.0225
A19	VC	spectral filtering	0.0317

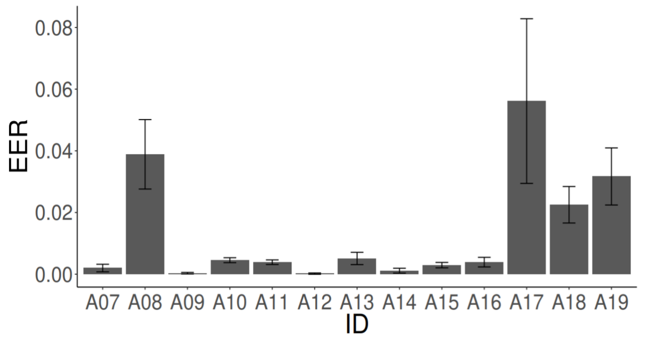


Fig. 5. The averaged EER for 13 attack types in evaluation set. The barplot indicate averaged EER with one standard deviation error bar.

IV. Related Work

Hadid et al. [20] studied general spoofing and anti-spoofing in the biometric authentication system. They described an evaluation methodology for assessing spoofing and discussed its countermeasures. [21]

Several techniques have been proposed to detect voice replay or synthesis attacks replayed through loudspeakers. Liu et al. [22] proposed the use of wearable devices, such as eyeglasses, earbuds, or necklaces, to detect voice liveness. They achieved approximately 97% accuracy in detecting voice liveness but relied on the use of earbuds. Zhang et al. [23] monitored unique articulatory gestures using sound wave reflection techniques to check voice liveness. They achieved approximately 99% accuracy but relied on users physically holding their devices near their ears. As part of the ASVspoof 2019 competition, several researchers have proposed machine learning-based liveness detection solutions. According to the reported recent competition results, the EERs varied from 0.22% to 92.36% for LA attacks, and 0.39% to 92.64% for PA attacks. However, the top five systems [6], [12]–[15] for both the LA and PA sets all used the ensemble approach and combined multiple models (see Table I). This is because the competition only emphasized the accuracy aspects of building a voice spoofing attack detection system, and did not consider the real-world deployment scenarios where minimizing model complexity and detection latency are also integral. To that end, most of the top performing solutions explored to date would not satisfy those model latency and complexity requirements. In comparison, the proposed ResMax models can perform competitively well even with only a single model and 262K parameters.

V. Discussions

A. Speech and Non-Speech Part Removal

If the quiet parts of the beginning and end of a given sample are noises that provide no information, training the model with only the speech part should improve the overall performance. To demonstrate this, we experimented with removing non-speech parts and training those processed samples. However, unlike our expectations, our models performed better when all parts of a given sample were used for training and testing. These results are inline with the results presented in [6]. They also trained their CQT-LCNN model after removing non-speech parts but demonstrated degrading performances. One possible explanation is that when a speaker is turned on, and even if there is no speech being replayed during the beginning or end of a given sample, some inherent noises generated through that speaker are being picked up and trained by our deep learning models.

B. Variability of the Model Performance

As a model is trained, the parameters are fitted in a way that optimizes the loss function using randomly selected batch samples during each epoch. Even if we train with the same number of epochs, the model parameters fit differently each time because the batch samples are picked randomly. Therefore, the EER and t-DCF results are also slightly different for each training session. We conducted 10 training sessions for each model to determine the variability of the results, and presented standard deviations around 0.19. Since there were 10 random variables, the t-test was used to evaluate whether the differences in the sample mean of the EER of the proposed models and the EERs of other models were statistically significant.

C. High Performing Model Architectures for Detecting Voice Spoofing

As can be seen from Table I, all the models that achieved excellent results in the ASVspoof 2019 competition used various deep learning methods in an ensemble, and in particular, ResNet and LCNN models were considerably used. The proposed ResMax is also an architecture that combines ResNet and LCNN, and it is shown that the deep learning architecture of ResNet and LCNN works well for voice spoofing detection.

VI. Conclusions

Existing voice spoofing attack detection solutions have been designed without considering real-world model complexity and detection latency requirements, and often consist of multiple heavy and complex deep learning models. Such solutions would not be considered suitable given the tight model size and latency requirements. In comparison, our CQT-1_120-ResMax model used only a single deep learning model with far fewer model parameters to outperform the top performing PA solution from evaluation set, achieving an EER of 0.37% compared to the current best competition EER of 0.39%, which is an ensemble solution. As for the LA set, we rank third with an EER of 2.19%, just behind the second best ensemble solution that achieved an EER of 1.86% in the evaluation set. Among the single model systems, although CQT-1_120-ResMax used the least number of parameters, it demonstrated significant superiority in detection accuracy.

Acknowledgment

This work was conducted at Samsung Research. The authors would like to thank Samsung Research Security Team for the helpful discussions. IK was supported by the National Research Foundation of Korea(NRF) grant funded by Ministry of Science and ICT (2020R1C1C1A01013020)

References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurou, "Tacotron: Towards end-to-end speech synthesis," in Proc. Interspeech 2017. Stockholm: ISCA, 2017, pp. 4006–4010.

[2] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in 6th International Conference on Learning Representations. Vancouver: ICLR, 2018.

[3] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilci, M. Sahidullah, and A. Sizov, "Asvspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in Proc. Interspeech 2015. Dresden: ISCA, 2015, pp. 2037–2041.

[4] 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. Stockholm: ISCA, 2017, pp. 2–6.

[5] 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. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2249

[6] 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. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1768

[7] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," in INTERSPEECH, 2017.

[8] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic Speaker verification," Computer Speech & Language, 20 February 2017, 02 2017. [Online]. Available: http://www.eurecom.fr/publication/5146

[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. Hyderabad: ISCA, 2018, pp. 681–685.

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

[13] 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," in Proc. Interspeech 2019, 2019, pp. 1023–1027. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1230

[11] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2884–2896, 2018.

[12] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-Spoofing with Squeeze-Excitation and Residual Networks," in Proc. Interspeech 2019, 2019, pp. 1013–1017. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1794

[14] Y. Yang, H. Wang, H. Dinkel, Z. Chen, S. Wang, Y. Qian, and K. Yu, "The SJTU Robust Anti-Spoofing System for the ASVspoof 2019 Challenge," in Proc. Interspeech 2019, 2019, pp. 1038–1042. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2170

[15] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble Models for Spoofing Detection in Automatic Speaker Verification," in Proc. Interspeech 2019, 2019, pp. 1018–1022. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2505

[16] C. Elkan, "The foundations of cost-sensitive learning," in Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01). Washington: IJCAI, 2001, pp. 973–978.

[17] 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," in Odyssey 2018 The Speaker and Language Recognition Workshop, 2018.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2014.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[20] A. Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometrics systems under spoofing attack: An evaluation methodology and lessons learned," IEEE Signal Processing Magazine, vol. 32, no. 5, pp. 20–30, 2015.

[21] S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans, Handbook of biometric anti-spoofing : Presentation attack detection. Editors: Marcel, S., Nixon, M.S., Fierrez, J., Evans, N. (Eds.); Springer International Publishing, 2018, 2nd ed.; ISBN: 978-3319926261, 09 2018. [Online]. Available: http://www.eurecom.fr/publication/5667

[22] R. Liu, C. Cornelius, R. Rawassizadeh, R. Peterson, and D. Kotz, "Vocal resonance: Using internal body voice for wearable authentication," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, pp. 1–23, 2018.

[23] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authenticatio," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Texas: ACM, 2017, pp. 57–71.