Speech Communication 141 (2022) 56–67

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

A study on data augmentation in voice anti-spoofing

Ariel Cohen [*,**], Inbal Rimon [**], Eran Aflalo, Haim H. Permuter

Ben Gurion University, Israel

ARTICLE INFO

Keywords:
ASVspoof 2021
Audio data augmentation
Data-centric AI
SpecAugment
Voice anti spoofing
Voice deep fake

ABSTRACT

In this paper we perform an in depth study of how data augmentation techniques improve synthetic or spoofed audio detection. Specifically, we propose methods to deal with channel variability, different audio compressions, different bandwidths and unseen spoofing attacks. These challenges, have all been shown to significantly degrade the performance of audio based systems and anti spoofing systems. Our results are based on the ASVspoof 2021 challenge, in the Logical Access (LA) and Deep Fake (DF) categories. Our study is *Data-Centric*, meaning that the models are fixed and we significantly improve the results by manipulating the data. We introduce two forms of data augmentation - compression augmentation for the DF part, and compression and channel augmentation for the LA part. In addition, we introduce a double sided log spectrogram feature design that improves the results significantly by centering the sub-bands of interest, where the discriminating spoofing artifacts can be localized. Furthermore, a new type of online data augmentation, SpecAverage, is introduced. This method includes masking the audio features with their average value in order to improve generalization. Our best single system and fusion schemes both achieve state of the art performance in the DF category, with an EER of 15.46% and 14.27%, respectively. Our best system for the LA task reduced the best baseline EER by 50% and the min t-DCF by 16%. Our techniques to deal with spoofed data from a wide variety of distributions can be replicated and can help anti spoofing and speech based systems enhance their results.

1. Introduction

The use of the human voice for tasks such as Automatic Speaker Verification (ASV), spreading news on social media, and communicating using digital devices has become very popular. ASV, for example, is used in many applications such as voice mail, telephone banking, call centers, biometric authentication, forensic applications and more.

Nowadays, generating synthetic speech has become a doable task, as many new algorithms emerge and technology advances. These algorithms include Text to Speech (TTS) (Dutoit, 1997), and Voice Conversion (VC) (converting speech from source speaker to target speaker) (Zhao et al., 2020; Kobayashi et al., 2021) among others. Spoofing is the process of creating synthetic speech where the goal is either to fool algorithm based solutions/automatic solutions or the human ear, by creating perceptually natural sounding speech that mimics a target speaker. Another form of spoofing can be physically replaying a recorded audio sample of a specific speaker. Research has shown that both audio technology and the human ear are vulnerable to voice spoofing. In the past few years, anti spoofing for ASV has become a field of interest in the research community; four bi-annual international challenges (Wu et al., 2015; Kinnunen et al., 2017; Todisco et al., 2019; Yamagishi et al., 2021) have been held in which the goal has been to improve the ability to discriminate bonafide speech from spoofed speech.

1.1. Practical anti-spoofing challenges

Aside from the challenges of detecting whether a given audio signal is bonafide or spoofed, practical anti spoofing systems face the following challenges.

1. **Compression:** Lossy audio compressions typically contain some form of non linear quantization together with selective frequency reduction. Compression may be a cause of audio quality degradation and transmission mismatch that can degrade the performance of audio systems such as ASV systems (Jarina et al., 2017), speaker recognition systems (Stauffer and Lawson, 2009), and anti spoofing systems. Common compressions are MP3 (Sterne, 2012), Advanced Audio Coding (AAC) (Bosi et al., 1997) and G.722 (Mermelstein, 1988), among others.

* Corresponding author.
E-mail addresses: ariel5@post.bgu.ac.il (A. Cohen), inbalri@post.bgu.ac.il (I. Rimon), eranaf@post.bgu.ac.il (E. Aflalo), haimp@bgu.ac.il (H.H. Permuter).
** The first two authors, Ariel and Inbal, have an equal contribution in this research.

https://doi.org/10.1016/j.specom.2022.04.005
Received 25 October 2021; Received in revised form 1 March 2022; Accepted 12 April 2022
Available online 29 April 2022

2. **Channel effects:** Transmitting compressed audio through a channel might induce transmission related data loss such as packet loss, noise and more. This type of data loss can degrade the performance of audio feature based systems, as stated in Besacier et al. (2003). Channels for example can be VoIP, landline, cellular or satellite.
3. **Bandwidth differences and filtering:** Audio codecs can differ by bandwidth, as some codecs are narrow band and others are wide band. In addition, some include band pass filtering prior to transmission, a fact that may cause information loss at high frequencies, which may contain crucial information necessary for detecting spoofing attacks (Tak et al., 2020a).
4. **Unseen spoof attacks:** One of the main challenges of an anti spoofing system is to generalize and to detect unseen attacks from an unknown distribution. In Zhang et al. (2021c), the authors performed a cross dataset study that included the VCC2020 (Zhao et al., 2020) dataset, among others, and showed significant degradation in performance.

1.2. Model-centric vs. Data-centric

An Artificial Intelligence (AI) system is typically composed of data and a model, which go hand in hand in producing the desired results. A normal optimization process consists of constantly improving the statistical model and the data in an iterative manner. While both are important, attention usually shifts towards one of the following approaches.

1. **Model-Centric Approach:** In this approach, the data is fixed and empirical tests are performed with respect to the model architecture and training procedure in order to maximize the results.
2. **Data-Centric Approach:** In this approach, the model is fixed and changes/improvements are constantly made in the data set in order to maximize the results.

Our study is mainly Data-Centric. We chose models that had displayed good performance on the ASVspoof 2019 data, and focused our efforts on data augmentation and feature design in order to tackle the challenges of ASVspoof 2021.

1.3. Motivation

In Fig. 1, we can see how the channel mismatches caused by compression, transmission effects, and bandwidth differences affect the score distribution of the Resnet model 2.1 with log spectrogram features. In this experiment, we trained the Resnet model using the original ASVspoof 2019 training set. Scores were produced on both the original ASVspoof 2019 development set, and a reference development set that underwent simulated transmission, possible packet loss and compression. Aside from the differences in the score range, we can see that the original development set is relatively separable (bonafide scores are mostly different from spoofed scores), whereas the transmitted development set is not separable. This led to a high Equal Error Rate (EER), both on our simulated transmitted data set and on the evaluation data set. This demonstrates the sensitivity of audio based systems to harsh changes in the channel, and the importance of channel related data augmentation. We encountered similar effects using compression augmentation without channel effects. These findings motivated our current work.

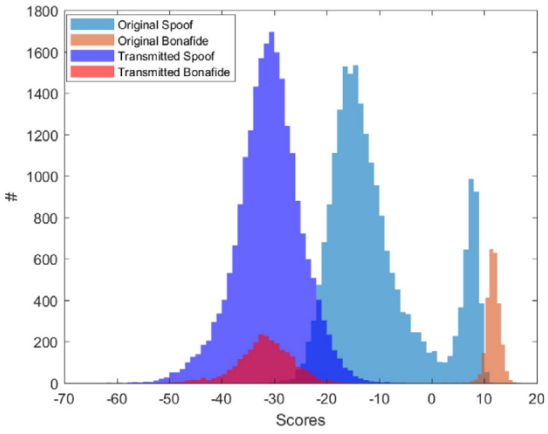

Fig. 1. Score histograms from ResNet trained on the original training data set and tested on two development sets (original and transmitted). Each data set is separated into spoof and bonafide. The original data scores (blue and orange) are quite separable, while the augmented data scores are completely overlapping and indistinguishable (purple and red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1.4. Compression and channel augmentation methods

Compression and channel augmentation are not new methods applied in the speech community, in the context of extensive efforts to make speech related systems, such as automatic speech recognition (ASR), work well in real life conditions. In Vu et al. (2019), the authors use known audio codec simulations in order to train telephony speech recognition systems. They classify the simulation methods into three main categories according to their distortion severity in terms of their spectrogram analysis. In Hailu et al. (2020), the authors improve the results and robustness of an end-to-end ASR model in four languages by expanding the training set. This was done by using different audio codecs at the raw speech level: audio codecs with a changed bit rate, sampling rate, and bit depth were considered. In Mayorga et al. (2003), the authors researched the effect of packet loss on speech recognition systems over IP connections. In the anti-spoofing domain, several augmentation techniques were also proposed. In Chen et al. (2021b), the authors proposed a compression augmentation pipeline that includes MP3 and AAC. In addition, a channel and codec augmentation pipeline, which includes device impulse response convolution to add robustness was proposed. In Chen et al. (2020), the authors simulated a call center environment by performing a playback of the ASVspoof 2019 data over voice calls and recording the received audio at the receiver's end. Using VoIP channel characteristics, a reduced bandwidth of 8 kHz, and Twilio's default OPUS codec, the authors achieved good generalization performance on the ASVspoof 2019 evaluation set. In Chen et al. (2021a), the authors used the same playback simulation together with reverberation and background noise in order to achieve good results in ASVspoof 2021.

1.5. Database and setup

The ASVspoof 2021 challenge contained two scenarios that are related to the issues stated above 1.1. Both scenarios contained bonafide and spoofed speech segments that had been generated with TTS and VC algorithms. In the Deep Fake (DF) category, new and never-seen-before spoofing methods had been used, and the audio files might have undergone compression (such as MP3, m4a and others) with various bit rates. In the Logical Access (LA) category, the audio files had been communicated across telephony and VoIP networks with various coding and transmission effects. Both scenarios contained unseen spoofing attacks. We used these two scenarios to benchmark our ideas. In order

to compare this work with the ASVspoof 2021 competition results, we only used the ASVspoof 2019 LA data set for training and development, as stated in the ASVspoof 2021 evaluation plan (Delgado et al., 2021). Training and development partitions were kept the same as in ASVspoof 2019 for both the DF and LA parts of the competition. In this paper, data that has not been augmented is referred to as original data.

1.6. Main contributions

In this paper we introduce several techniques that improve the robustness of anti spoofing systems to channel variability and compression, which are listed below.

1. We introduce compression data augmentation methods that improve anti spoofing system performance with compressed data.
2. We introduce channel robust data augmentation methods that improve anti spoofing system performance with compressed data that has been transmitted, filtered and down sampled.
3. We introduce a new feature design, double sided log spectrogram centering, which improves the learning process by re-allocating the sub-bands of interest to the center of the receptive field. We show how using this method improves the results significantly.
4. We introduce a new form of online data augmentation, SpecAverage, that generalizes the SpecAugment technique introduced in Park et al. (2019). In our experiments, SpecAverage showed good performance.

Our ideas were tested in the ASVspoof 2021 challenge in the DF and LA categories: We achieved state of the art performance in the DF category, both for our single system and for our system fusion. In addition, we also tested our ideas on the ASVspoof 2019 database and achieved strong results.

The rest of the paper is arranged as follows. Section 2 describes the deep learning models we used. In Section 3 we elaborate on the features we used and provide analysis regarding the effect that compression and transmission have on them. In Section 4 we introduce our new feature design. In Section 5 we present our data augmentation methods: compression augmentation (for DF) and channel augmentation (for LA). We then introduce SpecAverage. Sections 6 and 7 contain our results and analysis for the DF and LA parts. In Section 8 we test our ideas on the ASVspoof 2019 evaluation set and provide analysis. In Section 9 we discuss insights obtained from our work, and Section 10 contains conclusions and future work. Our augmentation methods are publicly available at: https://github.com/InbalRim/A-Study-On-Data-Augmentation-In-Voice-Anti-Spoofing.

2. Models

In this section we present the deep learning models we used: ResNet-34, SEnet and One Class Softmax (OCS) ResNet. The first two models were chosen for this research due to the fact that they are simple and well known, and have been used for the spoofing task with good results, while the third model was used since it showed good performance in the ASVspoof 2019 competition. All of the models are CNN based, which allows them to treat the spectrogram as if it were an image and thus capture the spatial relationship in time and frequency. While there are a lot of existing solutions, the top performing systems are based on CNNs together with the classical front-end and back-end partition. End-to-end systems are also being researched, but to the best of our knowledge do not achieve state of the art performance at this time.

2.1. ResNet

ResNet-34 is commonly used for image and audio tasks, as described in Dou et al. (2021) and He et al. (2016). The architecture we used is based on ResNet as outlined in Dou et al. (2021), with 2 main modifications based on experiments conducted prior to this work, as follows.

1. **Optimizer:** We used the AdamW (Loshchilov and Hutter, 2017) optimizer, which includes different parameters: Weight Decay (WD), β_1 and β_2. We chose WD $= 5 \cdot 10^{-8}$ and $\beta_1, \beta_2 = (0.81, 0.8991)$.
2. **Loss Function:** We experimented with the Binary Cross-Entropy (BCE) and Binary Focal-Loss (BFL) functions, and with different class weights. We got the best results using class weights where the bonafide class is weighted 10 times more than the spoof class with both loss functions. In addition, BCE provided slightly better results than BFL.

2.2. SENet

While in ResNet blocks the input channels are equally weighted, in SE blocks a different weight is given to each channel using Squeeze and Excitation (SE). As suggested in Dou et al. (2021), the squeeze is performed using average pooling. To simplify calculations, in order to perform excitation the input dimension is first reduced, followed by a ReLU activation, and then extended, followed by a sigmoid activation.

2.3. OCS-ResNet

In Zhang et al. (2021a), the authors presented a model based on ResNet-18 with an attentive pooling layer, and a one class softmax function as follows:

$$L_{OCS} = \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + e^{\alpha(m_{y_i} - \hat{w}_0\hat{x}_i)(-1)^{y_i}}\right) \tag{1}$$

where $m_0, m_1 \in [-1, 1], m_0 > m_1$ denote the angular margins between the classes, $\hat{w}_0$ denotes a normalized weight vector, $y_i \in \{0, 1\}$ denotes the class and $\hat{x}$ denotes the normalized vector of a target class embedding. We used this model with the same parameters as stated in the paper.

2.4. Model training

Each one of the three models was trained separately, In order to achieve minimal correlation and to maximize the fusion results. Different augmentation methods were used for each model, both for training and for development. All of the training and development data was chosen without any additional knowledge about the evaluation set, aside from what is stated in the evaluation plan. The specific composition of the augmented data sets is stated in the upcoming sections.

3. Audio processing and features

In this section we state the audio features we used with each model, including pre-processing done prior to feature extraction. We visualize and provide insights on how compressions, different channels and other conditions affect those audio features.

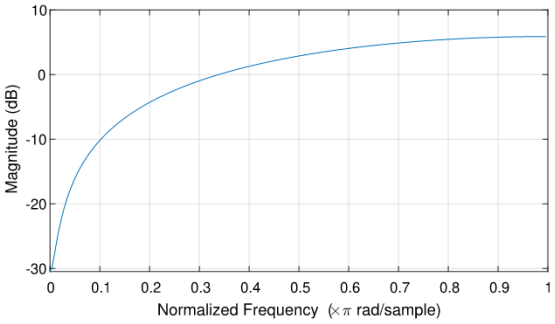

Fig. 2. Pre-emphasis filter frequency response.

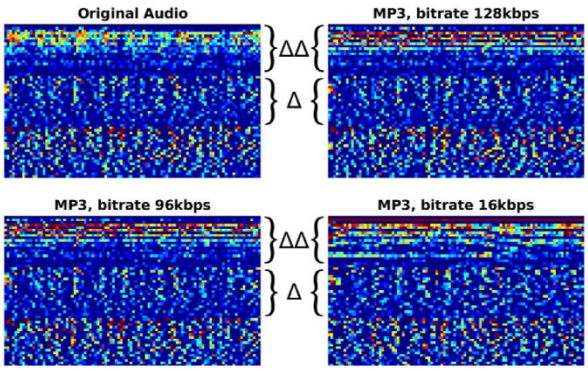

Fig. 3. LFCCs of compressed audio comparison. Compression affects the dynamic features (Δ and $\Delta\Delta$) significantly.

3.1. Motivation for chosen input features

Research performed in Sriskandaraja et al. (2016), Paul et al. (2017) and Tak et al. (2020a), showed that the most discriminating artifacts for spoof detection are localized in the high and low frequencies (0–1 kHz, 7–8 kHz). Due to that fact, we chose to use LFCC and log spectrogram based features. The log spectrogram is a high resolution feature containing all of the information for both significant sub-bands, while LFCC contains information for both sub-bands with an equal resolution. In addition, that can explain why the use of mel frequency scale filters along the speech bandwidth (MFCC) might not be the best feature choice for synthetic speech detection, even though it is the most common approach for front-end filter design in speaker recognition systems. Many of the high frequency artifacts are lost due to the low cepstral resolution in the upper band caused by the mel filter spacing.

3.2. Pre-emphasis

A common pre-processing tool used to compensate for the average spectral shape of a speech signal is pre-emphasis, which emphasizes higher frequencies. Typically, pre-emphasis is applied as a time-domain FIR filter with one free parameter. Unless specifically stated otherwise, pre-emphasis was used for all features using the following filter:

$$x[n] = x[n] - 0.97x[n-1] \tag{2}$$

This is a high pass filter which emphasizes the high frequencies, as shown in Fig. 2. The use of pre-emphasis serves two main purposes: first - to compensate for the average spectral shape, and second - to emphasize the higher frequencies where the discriminating information for spoof detection is located.

3.3. Log spectrogram (LogSpec)

The spectrogram of an audio signal was proven to be effective as a neural network input in Cheuk et al. (2020), and specifically for spoof detection in Lai et al. (2019). The spectrogram created using the STFT (short-time Fourier transform) is calculated as follows:

$$\text{LogSpec} = \log(|STFT(x)|^2) \tag{3}$$

where x is the audio signal. We used a frame length of 25 ms, a hop of 10 ms, and a total fixed length of 5 s. LogSpec was used with the SEnet and ResNet models.

3.4. Linear Frequency Cepstral Coefficients (LFCC)

We used LFCC with a window size of 20 ms and an overlap of 10 ms. We used both Δ and $\Delta\Delta$ as dynamic features. 20 coefficients were selected. The above resulted in a frequency dimension of 60, while the time dimension was fixed at 450. We used repeat padding for shorter utterances and randomly sliced the features out of longer utterances. LFCC was used with the OCS-ResNet model.

3.5. Compression effects on LFCC features

In Fig. 3 we can see four images of LFCCs of the same utterance. Processing included MP3 compression with various bitrates and then decompression back to 16 kHz and 16 bits per sample FLAC format, to match the data format stated in the competition evaluation plan. As seen in Fig. 3, MP3 compression affects the LFCC of a given input. While the static part (lower 20 coefficients) are affected mildly, the dynamic features (Δ and $\Delta\Delta$) are changed significantly, even between different bitrates of the same compression.

3.6. Compression and transmission effects on audio signals

In Fig. 4, we can see that for the same utterance, there are significant differences in the frequency response between the original audio file and processed versions of the original file that have undergone compression, simulated transmission, packet loss, filtering and down sampling. All of the above are clearly a source of channel mismatches that degrade performance, as seen in Table 5.

4. Feature design

In this section, we propose a new feature design for anti-spoofing that improved our results significantly. Our new feature design re-allocates the discriminating features located at the high and low frequencies of the spectrogram to the middle of the receptive field (Luo et al., 2016), giving them a larger impact on the learning process. We explore re-allocating both high and low frequencies, and in addition test the effects of using pre-emphasis. Finally, we elaborate on our online normalization methods.

4.1. Motivation

Much work has been invested in finding which part of the spectrum is more relevant for the spoof detection task. In Sriskandaraja et al. (2016), a sub-band analysis was performed based on the anti spoofing (SAS) corpus. The authors showed that the sub-bands containing the most discriminating information are 0–1 kHz and 7–8 kHz. More evidence that high frequencies contain discriminating information was shown in Paul et al. (2017). In Tak et al. (2020a), an in depth sub-band analysis was performed, in which both CQCC and LFCC features were tested with respect to the ASVspoof 2019 data base and a Gaussian mixture model (GMM) classifier. The analysis shows that spoofing attacks have different artifacts that can be highly localized to high and low frequencies. This motivated us to search for an appropriate feature design that would be more efficient in both capturing these artifacts and learning from them.

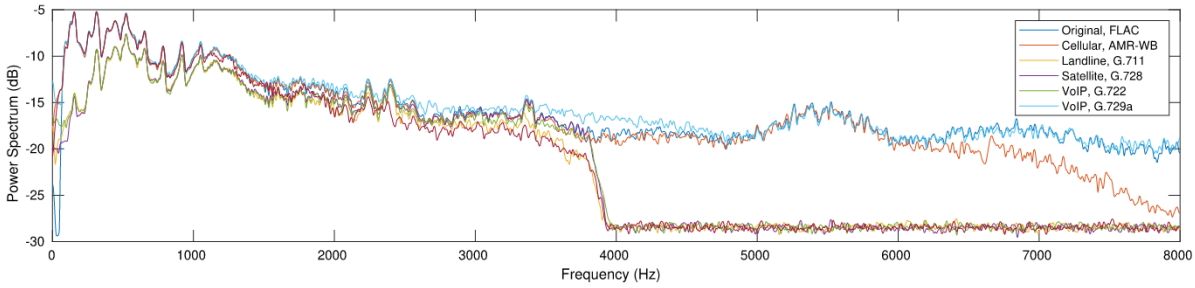

Fig. 4. Power spectrum of the same utterance, augmented with different transmissions and codecs. Differences are significant in the lower band (0 kHz–1 kHz), where speech frequencies are located, mid band (3.5 kHz–4.5 kHz), where the effect of down sampling can be seen, and upper band (7 kHz–8 kHz). All emphasize the channel variation.

Table 1
Double sided LogSpec comparison, ResNet. The best results are achieved by using pre-emphasis and centering the high frequencies, or centering the low frequencies without using pre-emphasis. The frequency resolution is constant and equal to 512 in all experiments.

Double sided	Size	Frequencies centered	Pre-emphasis	DF EER
✗	257 × 500	–	✓	25.24
✓	512 × 500	High	✓	**18.81**
✓	512 × 500	Low	✓	24.46
✗	257 × 500	–	✗	20.7
✓	512 × 500	High	✗	19.10
✓	512 × 500	Low	✗	**18.46**

In addition, recent work regarding the receptive field in CNNs, showed that the artifacts located in the center of the input have a larger weight in the learning process (Luo et al., 2016). During training, these artifacts propagate through a larger number of paths within the network and thus have a larger weight on the gradients calculated. This makes them more significant to the learning process. In this paper, we propose a feature design that, by centering the areas where spoofing artifacts are most likely to be located, enables the models to learn better by focusing on the regions of impact.

4.2. Double sided log spectrogram

Due to the fact that audio is a real signal, the LogSpec is symmetrical in frequency. Hence, it seems that using the double sided LogSpec as a feature may not be useful, as it contains redundant data and might just add run time. As a matter of fact, most common software packages return a one sided log spectrogram as a default. Despite that fact, motivated by the research conducted on the receptive field in CNNs, we decided to test the use of double sided spectrograms in order to center either the high or the low frequencies (which contain the discriminating artifacts caused by the spoofing process) to the middle of the receptive field. We conducted several experiments using the ResNet model. We experimented centering both high and low frequencies, with and without pre-emphasis. The evaluation set is the DF evaluation set, which contains unseen spoof attacks, unseen compression methods and a variety of unknown bitrates. The frequency resolution used depends on the number of frequency bins/DFT points, nfft, and is equal in all cases. nfft is typically chosen with respect to the frame length and sampling rate:

$$\text{nfft} = 2^{\lceil \log(frame\_length \cdot fs) \rceil} \tag{4}$$

where fs is the sampling frequency (16 kHz in our case) and the frame length is 25 ms. In the case where the spectrogram is one sided - half of the frequency bins are removed due to the symmetry (excluding the DC frequency bin). The results of the experiment are shown in Table 1, and a visualization is shown in Fig. 5.

As seen in Table 1, we provide a few conclusions listed below.

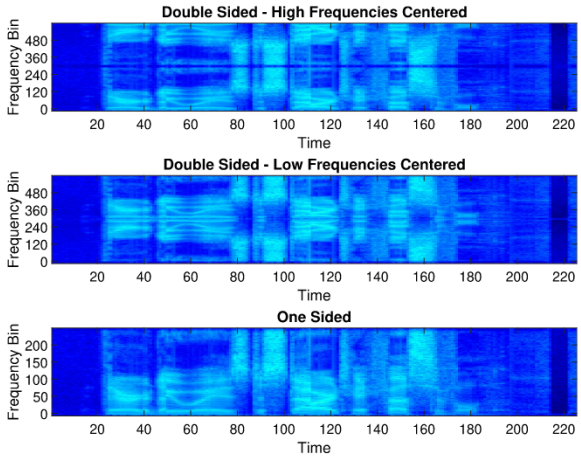

Fig. 5. Double sided and one sided LogSpec feature design. While no new information is added, the performance improves significantly when the discriminative information in high and low frequencies is centered.

1. Using ResNet with the double sided LogSpec, pre-emphasis and high frequency centering reduced the DF evaluation EER by 25.45% with respect to the one sided experiment (with pre-emphasis).
2. Using ResNet with the double sided LogSpec, no pre-emphasis and low frequency centering reduced the DF evaluation EER by 11% with respect to the one sided experiment (without pre-emphasis).
3. Pre-emphasis is important for high frequency centering, since it emphasizes the high frequencies that contain the spoofing artifacts.
4. When using low frequency centering, the results were better without pre-emphasis. This makes sense since pre-emphasis does weaken the lower frequency band.
5. The reason for the significant improvements is the allocation of the high and low sub-bands to the center of the receptive field, using the double sided LogSpec.

4.3. Feature normalization

One of the methods that displayed impressive empirical performances in the DF category was feature (LFCC/LogSpec) normalization. We experimented with 3 different types of normalizations:

$$\mathcal{N}_{\text{min-max}}(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{5}$$

$$\mathcal{N}_{\text{mean}}(x) = \frac{x - \text{mean(x)}}{\max(x) - \min(x)} \tag{6}$$

$$\mathcal{N}_{\text{standard}}(x) = \frac{x - \text{mean(x)}}{\text{std}(x)} \tag{7}$$

Table 2
DF compression data set.

Bitrate [kbps]	MP3	AAC/m4a
16	✓	
48	✓	
64		✓
96	✓	✓
128	✓	✓
160	✓	

Table 3
EER performance on compressed data (DF data set) with and without augmentation in the training and development sets.

System	Augmentation	DF Eval EER
OCS-resnet	✗	29.31
OCS-resnet	✓	**28.52**
Resnet	✗	46.49
Resnet	✓	**17.51**
SEnet	✗	40.32
SEnet	✓	**19.47**

where x denotes the feature matrix. All normalization methods are performed per individual sample, online (during training and testing). The computational price is low, as neither of the methods above affected our run time. We achieved the best performance in all systems using Eq. (5), where a simple linear projection mapped the features to $[0, 1]$. The $\mathcal{N}_{\text{min-max}}$ normalization reduced the OCS-ResNet DF EER by 18%, and ResNets DF EER by 7%, as stated in Table 9.

5. Data augmentation techniques

In this section we introduce two forms of offline augmentation techniques that were used to increase the performance of the anti spoofing systems with compressed data and channel variation. In addition, we will introduce SpecAverage, a new form of online data augmentation.

5.1. Compression augmentation

The augmentation was performed as follows:

1. An audio file was chosen;
2. A compression method was chosen out of: MP3 and AAC/m4a;
3. A bitrate was chosen;
4. The audio file was compressed with the chosen method and bitrate.
5. The compressed audio file was de-compressed back to the FLAC format, 16 kHz, 16 bits per sample.

The steps are performed to comply with the evaluation plan of ASVspoof 2021, in the DF part. We used the Pydub (Robert et al., 2018) and FFmpeg (Tomar, 2006) packages in order to first read the audio files and then augment them. Table 2 contains the augmented part of our DF dataset. MP3 and AAC/M4a codecs were chosen since they were stated in the evaluation plan. The bitrates were chosen based on standard usages for each compression method, including high quality and low quality. This augmentation method was performed offline, since performing it online and then extracting LFCC/LogSpec features was incredibly costly in CPU usage due to the feature extraction (and not due to the augmentation).

In Table 3 we see the effect that compression augmentation had on our models. A significant decrease in the EER can be seen after applying data augmentation.

Table 4
LA data set channels and codecs used.

Channel	Landline	Cellular	VoIP	Satellite
Codecs	G.711	AMR	Silk	G.728
	G.726	AMRWB	G.722	
		GSM	SilkWB	
			G.729	

Table 5
EER, min t-DCF performance on compressed data (LA data set) with and without augmentation.

System	Augmentation	Eval min t-DCF	Eval EER
OCS-resnet	✗	0.7500	21.41
OCS-resnet	✓	**0.3639**	**6.59**
Resnet	✗	0.9032	40.73
Resnet	✓	**0.2931**	**5.18**
SEnet	✗	0.9696	38.20
SEnet	✓	**0.2961**	**6.14**

5.2. Codec, channel effect, bandwidth difference augmentation

In order to perform this augmentation we used the audio degradation simulator in Ferras et al. (2016) with some adjustments to augment the training and development data as follows:

1. An audio file was selected;
2. For each channel (landline, cellular, VoIP or satellite) a random codec out of the list in Table 4 was chosen;
3. RMS normalization was performed in order to simulate transmission gain changes and normalization values were chosen from a uniform distribution from [−30,−10] in dB;
4. Downsampling and band pass filtering were performed depending on the chosen codec;
5. The audio file was compressed according to the chosen codec, with a random bitrate;
6. Random packet loss was simulated;
7. The audio file was re sampled to either 8 kHz or 16 kHz, depending on the data set we wanted to create.

This type of augmentation was tested on the LA part. The channels and codecs we used are presented in Table 4. Augmenting the data in this way, we kept the original training and development partitions, and created a data set for down sampled data (8 kHz), regular data (16 kHz) and wide band codec data (16 kHz), as stated in Table 12. In Table 5, we can see that our augmentation improved the EER and min t-DCF (Kinnunen et al., 2020) significantly. All three models are tested on the LA evaluation data, with and without augmentation.

5.3. Augmentation differences

While other work has been done within speech related tasks to increase the robustness of systems using compressions and channel simulations, in this work there are a few fundamental differences, as listed below:

1. Down sampling and band pass filtering are performed, in accordance with the specific codec (not for all of the data);
2. All data is compressed and then decompressed back to the FLAC format, at 16 kHz with 16 bits per sample;
3. We use RMS normalization chosen from a uniform distribution [−30,−10] in order to simulate transmission gain changes;
4. No noise is added. This includes white noise, background noise, and foreground noise;
5. No additional impulse responses are used (for example for reverberation simulation, microphone variation);
6. Our compression augmentation pipeline supports all FFmpeg compressions, including MP3, AAC, Opus, Vorbis, and others.

Table 6
Augmentation policy. m_f and m_t denote the number of frequency and time masks, respectively. T and F represent the number of time values or frequency bins masked. In the case of SpecAugment, the features have been normalized to have zero mean.

Policy	Method	Feature	m_f	m_t	T	F
None	–	–	0	0	0	0
SAv1	SpecAverage	LFCC	1	0	0	12
FAu1	FreqAugment	LFCC	1	0	0	12
SAu1	SpecAugment	LFCC	1	0	0	12
SAv2	SpecAverage	LFCC	1	1	80	12
SAv3	SpecAverage	LogSpec	0	1	10	0
FAu3	FreqAugment	LogSpec	0	1	10	0
SAu3	SpecAugment	LogSpec	0	1	10	0
SAv4	SpecAverage	LogSpec	1	0	0	10
FAu4	FreqAugment	LogSpec	1	0	0	10
SAu4	SpecAugment	LogSpec	1	0	0	10

5.4. SpecAverage

Replacing random blocks of the input feature with a constant value during the training process (masking) has been shown to force deep neural networks to be more robust and to yield improved performance, both in speaker recognition and in anti-spoofing. In Park et al. (2019), the authors introduce SpecAugment, a technique that includes masking log mel spectrograms that are normalized to have a zero mean. In Chen et al. (2020), the authors introduce FreqAugment, where log filter banks are masked using the value 0, regardless of their mean value (normalization is performed on the utterance level). Here, we introduce SpecAverage, a generalized variant of SpecAugment, for the situation where the features do not have 0 as a mean value. In our experiments, SpecAverage has shown better performance then FreqAugment. We first define general parameters and then elaborate with regard to augmentation policies that have been used both on LFCC and on LogSpec features, as listed below:

1. **Frequency masking:** f consecutive frequency bins or coefficients $[f_0, f_0 + f)$ are masked with a constant value, where f is chosen from a uniform distribution from 0 to the frequency mask parameter F, and f_0 is chosen from $[0, \nu - f)$. ν is the number of frequency bins or coefficients;
2. **Time masking:** t consecutive time steps $[t_0, t_0 + t)$ are masked with a constant value, where t is first chosen from a uniform distribution from 0 to the time mask parameter T, and t_0 is chosen from $[0, \tau - t)$. τ is the total number of time steps;
3. **FreqAugment:** masking with the value 0;
4. **SpecAugment:** masking features that have been normalized to have a zero mean with their mean value, 0;
5. **SpecAverage:** masking with the average feature value calculated online for each feature.

Keeping the same notation as in Park et al. (2019), Table 6 describes the training policies used.

5.4.1. Comparative study

For mel-spectrograms, the mean value might be 0 due to mean normalization. In this study, our features are not normalized to have a zero mean, and thus the mean value and 0 are distinct. A histogram of the average values for both features across the training set is presented in Fig. 6.

An interesting question that arose from the comparison was: is there a difference between SpecAugment and SpecAverage? In order to test that, we conducted a few experiments. In Table 7 we display a comparison between masking with the value 0 (FreqAugment), normalizing the features to have zero mean and then masking with the value 0 (SpecAugment), and masking with the mean value (SpecAverage). We can see that SpecAverage has boosted the performance of both ResNet and OCS-resnet with 2 different features, with respect to both

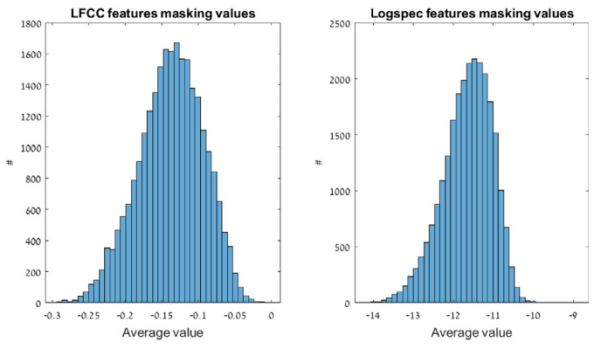

Fig. 6. Histograms of the average feature values across the training set.

Table 7
Performance comparison: SpecAverage, SpecAugment and FreqAugment on the DF evaluation. According to the experiments we conducted, SpecAverage showed better performance.

Feature	Model	Policy	DF Eval EER
LFCC	OCS-resnet	None	21.60
LFCC	OCS-resnet	FAu1	24.74
LFCC	OCS-resnet	SAu1	23.25
LFCC	OCS-resnet	SAv1	**21.51**
LogSpec	ResNet	None	17.03
LogSpec	ResNet	FAu3	20.55
LogSpec	ResNet	SAu3	21.28
LogSpec	ResNet	SAv3	**16.0**
LogSpec	ResNet	FAu4	16.6
LogSpec	ResNet	SAu4	17.58
LogSpec	ResNet	SAv4	**15.46**

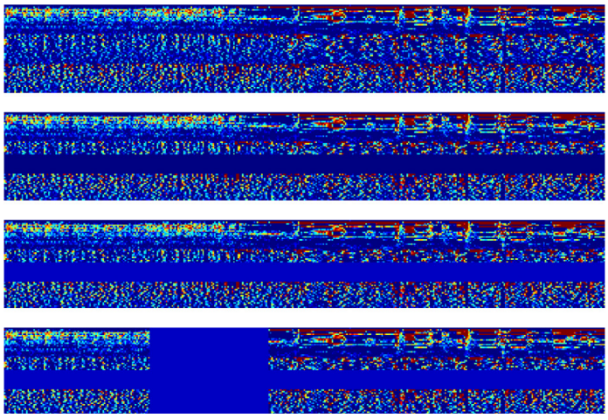

Fig. 7. LFCCs of utterances: from top to bottom - no masking, FreqAugment masking in frequency, SpecAverage masking in frequency, SpecAverage masking in time and frequency.

FreqAugment and SpecAugment. It is important to note that, in the case that the features are normalized to have zero mean, SpecAverage is equivalent to SpecAugment. In this sense, SpecAverage generalizes SpecAugment to a non-zero mean situation (see Fig. 7).

6. Deep fake: Results & analysis

In this section we present our results for our compression robust systems. We first present the data sets used, and then the results, followed by analysis.

6.1. DF data sets

During evaluation, we aimed to create a diverse data set for training and for development, while keeping the original partition of training

Table 8
Training and development data sets, DF.

Refr.	Orig	MP3					m4a		
		16	48	96	128	160	64	96	128
Tr1	✓	✓					✓		
Tr2	✓		✓				✓		
Tr3	✓		✓						
Tr4	✓			✓	✓	✓		✓	✓
Tr5	✓								
Dv1	✓		✓	✓	✓		✓	✓	✓
Dv2	✓	✓	✓		✓		✓		
Dv3	✓								

Table 9
Single system results, DF. The bottom three entries are the top competition results stated in Yamagishi et al. (2021). DSL-H and DSL-L stand for Double Sided LogSpec, with H for high frequencies centered and L for low frequencies centered. For L, pre-emphasis was not used.

Refr.	Features	Model	Data, Policy	Norm	Dev EER	Eval EER
Base-line	Audio	RawNet	–	✗	–	22.38
	LFCC	LCNN	–	✗	–	23.48
	LFCC	GMM	–	✗	–	25.25
	CQCC	GMM	–	✗	–	25.56
D1	LFCC	OCS	Tr4	✗	3.76	28.52
D2	LFCC	OCS	Tr4	✓	2.85	21.60
D3	LFCC	OCS	Tr4, SAv2	✓	2.92	21.94
D4	LFCC	OCS	Tr4, SAv1	✓	2.77	**21.51**
D5	DSL-H	ResNet	Tr1	✗	0.95	18.81
D6	DSL-H	ResNet	Tr2	✗	0.97	18.21
D7	DSL-H	SENet	Tr3	✗	1.22	19.47
D8	DSL-H	ResNet	Tr1	✓	0.88	17.03
D9	DSL-H	ResNet	Tr1,SAv3	✓	0.70	16.0
D10	DSL-H	ResNet	Tr1,SAv4	✓	0.48	**15.46**
D11	DSL-L	ResNet	Tr1,SAv3	✓	0.91	19.70
D12	DSL-L	ResNet	Tr1,SAv4	✓	0.49	**15.51**
T23	–	–	–	–	–	15.64
T20	–	–	–	–	–	16.05
T08	–	–	–	–	–	18.3

and development as in the ASVspoof 2019 competition. We chose the training set so that it contained low quality and high quality augmented audio with different compressions, and we chose the development set in such a way that it would contain unseen bit rates. We trained three different models with different features so we could reduce the correlation between them and then maximize fusion results. Table 8 contains the training and development data sets we used in the DF category.

Dv1 was used for the LFCC based models and Dv2 was used for the LogSpec models.

6.2. DF results

Our results for single systems are presented in Table 9.
It can be seen that:

1. Feature normalization shows significant improvement in both models - EER reduction of 24% in OCS-resnet, and 10% in ResNet;
2. SpecAverage improves the results of both models;
3. Our best single system, D10, used double sided Logspec, compression augmentation, feature normalization and SpecAverage, and achieves state of the art performance;
4. Using DSL-L, state of the art performance is achieved as well (D12).

6.2.1. DF fusion scheme

Our score level fusion scheme includes using the best two systems, which are D10 and D12. Fusion was performed using the mean of the best two systems, as stated in Table 10. A detection error tradeoff (DET) curve is presented in Fig. 8.

Table 10
System fusion, DF.

Single systems	Method	Eval EER
D10+D12	Mean	**14.27**

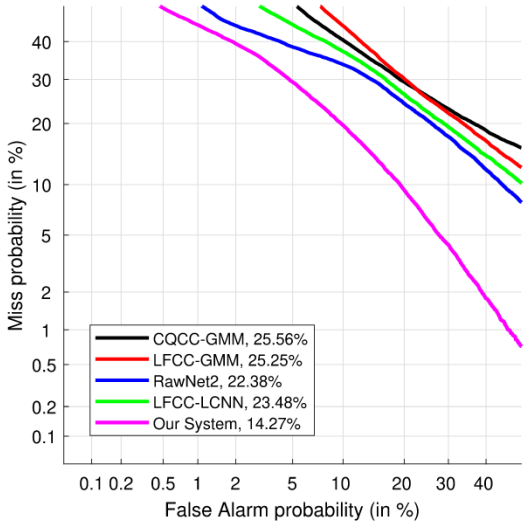

Fig. 8. DF DET curve.

Table 11
DF evaluation data conditions. VBR denotes the variable bit rate range. DF-C8 and DF-C9 have been compressed twice.

Cond.	Compression (Quality)	VBR [kbps]	Double compression
DF-C1	None	None	✗
DF-C2	MP3 (low)	80–120	✗
DF-C3	MP3 (high)	220–260	✗
DF-C4	m4a (low)	20–32	✗
DF-C5	m4a (high)	96–112	✗
DF-C6	ogg (low)	80–96	✗
DF-C7	ogg (high)	256–320	✗
DF-C8	MP3 (low) → m4a (high)	80–120, 96–112	✓
DF-C9	ogg (low) → m4a (high)	80–96, 96–112	✓

6.3. DF analysis

In this subsection we provide analysis regarding the sub-conditions of the DF dataset. Table 11 contains the DF evaluation conditions. Table 14 presents our system's EER for each condition. Our best system outperforms the baseline systems in all conditions. DF-C4 was the condition with the worst performance. This can be explained by the fact that DF-C4 is compressed using m4a with a very low bit rate, and we assume that the low quality created this performance bias. Surprisingly, the best performance was achieved in DF-C6, compressed using low quality ogg. This is a compression method that our model had not seen in training or development, emphasizing the compression robustness our system offers. Another interesting fact is that our system performed well in DF-C8. Compressing an audio file twice using different compression methods with different bit rates enlarges the information loss, and our system performed relatively well in that case.

7. Logical access: Results & analysis

In this section we present our results for the LA category. We first present the data sets used, and then the results, followed by analysis. Results are provided in terms of EER and min t-DCF (referred to as t-DCF), as stated in the ASVspoof 2021 evaluation plan.

Table 12
LA data sets.

Reference	Sample rate [kHz]	Codecs
1	8	All
2	16	All
3	16	AMR-WB, Silk-WB, G.722

Table 13
Single systems, LA.

Refr.	Features	Model	Train set	BR kHz	Eval EER	Eval t-DCF
Base-line	Audio	RawNet	–	16	9.50	0.425
	LFCC	LCNN	–	8	9.26	0.344
	LFCC	GMM	–	8	19.30	0.575
	CQCC	GMM	–	8	15.62	0.497
L1	LogSpec	ResNet	2	16	**5.18**	**0.293**
L2	LogSpec	SENet	2	16	6.14	0.296
L3	LFCC	OCSresnet	1	8	7.22	0.333
L4	LFCC	OCSresnet	1	8	6.65	0.343
L5	LFCC	OCSresnet	1, FAu1	8	6.59	0.363
L6	LFCC	OCSresnet	1, SAv1	8	**5.99**	**0.323**
L7	LFCC	OCSresnet	1,3	Both	6.99	0.348

7.1. LA data sets

In Table 12 we present the data sets we used in the LA partition. For each type of feature we needed to decide what sampling rate to use, since some of the codecs include filtering and down sampling to 8 kHz (narrow band codecs) and some were wide band codecs that were 16 kHz. For Logspec we used data that was resampled to 16 kHz regardless of the codec (resulting in 'half empty' spectrograms for the narrow band codecs). For LFCC, we either downsampled all of the data to 8 kHz and then performed feature extraction, or we performed effective bandwidth detection (using the obw() function in Matlab) and then trained 2 different models in order to achieve a band selective model, meaning that each new sample had its effective bandwidth calculated and then sent to the appropriate model (L7 in Table 13).

7.2. LA results

Our results for single systems are presented in Table 13. It can be seen that:

1. All of our single systems are trained with compression and transmission augmentation and perform better than the baseline systems;
2. SpecAverage improves the results of OCS-resnet and outperforms FreqAugment;
3. Our best single system, L1, reduced the best baseline EER by 44% and the best baseline min t-DCF by 15%.

7.2.1. LA fusion scheme

In the LA task, a weighted mean based on a grid search was performed. The development set contained all channels, all codecs and all conditions.
As seen in Table 15, the fusion scheme further improved the results. The best baseline system EER was reduced by as much as 50%, and the min t-DCF was reduced by 16%.

7.3. LA analysis

Table 16 contains the LA evaluation conditions.
Aside from LA-C1, LA-C4 and LA-C6, the codecs used are ones our model has not been trained on. We can see that there are different transmission settings as well. Table 17 shows the EER performance for all conditions. It can be seen that our system surpasses the baseline EER performance by a large margin. In addition, the performance difference between narrow band data (C2,C3,C5,C6) and wide band data (C1,C4,C7) is significant. We hypothesize that the removal of the high frequencies caused a clear degradation in performance. Finally, it can seen that our system performs relatively well even on unseen channels and codecs, as even the worst EER value (5.82) is low compared to the baseline equivalents.

8. Cross data set results

In this section we test the effectiveness of our ideas on a different data base, using the ResNet model. We first present our results on the ASVspoof 2019 evaluation set, and then present a comparison to other state of the art single systems.

8.1. Performance on ASVspoof 2019 evaluation set

Ablation analysis was performed on the ASVspoof 2019 evaluation set, the results are presented in Table 18. The training and development data sets we used are Tr1, Dv2 where we used compression augmentation for training, and Tr5, Dv3 in case where we did not. The evaluation set used was not augmented by compressions in both cases. DSL-H represents a double sided logspec with high frequencies centered. DSL-L represents the case where low frequencies were centered. Pre-emphasis was performed for one sided and DSL-H feature designs.

We can see that:

1. Using our double sided feature design increased the performance on this data set as well, in all cases;
2. Using SpecAverage increased the performance - in some cases moderately and in some cases significantly;
3. Centering the high frequencies eventually yielded better results than centering the low frequencies;
4. Using compression augmentation increased the performance as well;
5. Using all of our methods together (equivalent to system D10) resulted in an EER of 1.61%, which is an 88% reduction of the original EER for the system and is an overall strong performance for a single system on this data set.

8.2. Comparison of top performing single systems

Table 19 displays a performance comparison of top performing single systems. Our single system has strong performance with respect to other systems that use different front-end features.

9. Discussion and insights

In this section we will discuss interesting points and insights that we found during our research.

9.1. The importance of augmentation

Throughout this work, two different augmentation methods were proposed: compression augmentation for the DF part, and codec, channel effect, and bandwidth difference augmentation for the LA part. We have shown that both methods are crucial for the models to function properly with data that has witnessed these effects, or similar ones. Both augmentation types not only help models deal with the trained data, but render the models robust to different kinds of compressions and channels. The evidence of this is clear, as our best DF system has the best result on an unseen compression (ogg), and our best LA system has good results on unseen channels.

Table 14
DF evaluation results for the different conditions.

Refr.	DF-C1	DF-C2	DF-C3	DF-C4	DF-C5	DF-C6	DF-C7	DF-C8	DF-C9	EER
RawNet	26.98	27.63	27.49	26.72	27.23	18.80	18.67	18.74	19.10	22.38
LCNN	23.19	34.21	23.88	25.22	23.85	19.06	17.10	28.35	18.54	23.48
GMM LFCC	17.39	39.20	17.97	20.95	21.43	22.16	14.83	39.03	26.65	25.25
GMM CQCC	19.48	48.86	20.37	19.55	20.27	17.92	14.42	49.41	17.39	25.56
D10+D12	**14.45**	**15.53**	**14.31**	**19.32**	**14.80**	**11.17**	**12.47**	**12.12**	**15.95**	**14.27**

Table 15
Fusion systems, LA.

Single systems	Method	Eval EER	Eval t-DCF
L1-L7	Weighted mean	**4.66**	**0.2882**

Table 16
LA evaluation conditions.

Cond.	Codec	Bandwidth	Transmission
LA-C1	None	16	None
LA-C2	a-law	8	VoIP
LA-C3	unk.+μ-law	8	PSTN+VoIP
LA-C4	G.722	16	VoIP
LA-C5	μ-law	8	VoIP
LA-C6	GSM	8	VoIP
LA-C7	OPUS	16	VoIP

Table 17
LA evaluation results for the different conditions.

Refr.	LA-C1	LA-C2	LA-C3	LA-C4	LA-C5	LA-C6	LA-C7	EER
RawNet	5.84	6.59	16.72	6.41	6.33	10.66	7.95	9.50
LCNN	6.71	8.89	12.02	6.34	9.25	11.00	6.66	9.26
GMM LFCC	12.72	21.21	35.55	15.28	18.76	18.46	12.73	19.30
GMM CQCC	10.57	14.76	20.58	11.61	13.58	14.01	11.21	15.62
L1-L7	**3.03**	**5.04**	**5.82**	**3.21**	**4.80**	**5.82**	**4.29**	**4.66**

Table 18
Ablation analysis on ASVspoof 2019 evaluation set, ResNet. Using our augmentations and feature design results in strong performance on this data set as well. DSL-H and DSL-L stand for Double Sided LogSpec, with H for high frequencies centered and L for low frequencies centered. For L, pre-emphasis was not used.

Feature design	Trained with compressions	SpecAverage	EER
DSL-H	✓	✓	**1.61**
DSL-H	✓	✗	4.44
DSL-H	✗	✓	6.81
DSL-H	✗	✗	11.99
DSL-L	✓	✓	3.30
DSL-L	✓	✗	4.97
DSL-L	✗	✓	6.86
DSL-L	✗	✗	8.25
One sided	✓	✓	6.55
One sided	✓	✗	11.13
One sided	✗	✓	8.74
One sided	✗	✗	13.64

9.2. Online masking methods

Throughout this work, three different online masking methods were considered. Based on the tests that we performed we hypothesize that the reason SpecAverage had better performance is that the average value is statistically meaningful relative to the input feature, offering more information in addition to the regularization effect provided by the masking. It is important to note that time warping, one of the features in SpecAugment, has not been used as it is costly in computing resources. Despite that fact, masking with the average value gave consistently better results than masking with the value 0 (FreqAugment), or performing zero mean normalization and then masking with the value 0.

9.3. LFCC vs. Log spectrogram

We can see that the LFCC based model underperformed both of the LogSpec models. We believe that a possible explanation for this might lie in the dynamic LFCC features. As we visualized in Fig. 3, the differences between the same audio file uncompressed or compressed with different bitrates can be clearly seen, especially in the high frequency range. While the LogSpec contains all of the information as it comes, we believe that an additional amount of noise is produced during the LFCC extraction process while computing the high frequency derivatives. The differences in the dynamic features between bitrates can be explained by the fact that lossy compression of an audio signal involves removing high frequencies that are not heard by the human ear, and the lower the quality the more information is lost, together with harsher quantization. Future work could investigate using only parts of the dynamic LFCC features, or even removing them.

9.4. Dealing with unseen and double compressions

In the DF part, we encountered an interesting phenomenon: our model that has been trained on MP3 and m4a compressions performed best on low quality ogg, which is a compression method that had not been learned by the model. We believe this could be because, although the compression methods are different, they do remove specific frequencies that are based on human hearing perception tests, and in that manner they are similar. We believe that this fact contributed to this result. To further test this interesting fact we conducted an experiment, as seen in Table 20. The results were consistent.

9.5. The bitrate effect

We experimented with a large variety of bitrates. One of the challenges was to decide which bitrates to use for training and development for each compression method and which not to use. In order to further understand how to decide, we performed internal tests and came up with a few conclusions:

1. Knowing the exact bitrates in the test set gives the best performance (given that they are, in fact, known);
2. If the bitrates in the test set are not known, the best performance was achieved by using at least one low bitrate and at least one high bitrate—the cost of using more is time (training and testing), and does not always help;
3. The development set used in the training process should contain bitrates unseen during training.

10. Conclusions and future work

In this paper we performed an in depth study of how data augmentation affects voice anti spoofing systems. We presented two different types of data augmentation (for DF and for LA), that both significantly enhanced the results of the models we used. Our methods showed improvement in tasks that involve new spoofing attacks that have not been seen during training, compressed data, transmitted data and data with different bandwidths. We introduced a new form of feature design, double sided LogSpec centering, that by re-allocating the subbands of interest in the center of the receptive field, increased the

Table 19
Performance comparison of top performing single systems on the ASVspoof 2019 evaluation part. Our system uses a simple model, and by using our feature design we achieved strong performance.

System	Front-end feature	EER
LCNN (Lavrentyeva et al., 2019)	LFCC	5.06
ResNet18-GAT-T (Tak et al., 2021)	LFB	4.71
ResNet (Yang et al., 2021)	CQT-MMPS	3.72
GMM (Tak et al., 2020b)	LFCC	3.50
LCNN+CE (Das et al., 2021)	DASC-CQT	3.13
SE-Res2Net50 (Li et al., 2021a)	CQT	2.50
Resnet18-OC-softmax (Zhang et al., 2021a)	LFCC	2.19
Capsule network (Luo et al., 2021)	LFCC	1.97
LCNN-LSTM-sum (Wang and Yamagishi, 2021)	LFCC	1.92
MCG-Res2Net50+CE (Li et al., 2021b)	CQT	1.78
Res-TSSDNet (Hua et al., 2021)	Raw-audio	1.64
D10 (ours)	Double sided Logspec	**1.61**

Table 20
Compression robustness test. Our model performs well even on unseen compressions. The test data set used for comparison was the ASVspoof 2019 LA evaluation set.

System	Compression trained on	Compression tested on	EER
D4	MP3, M4a	MP3, M4a	3.63
D4	MP3, M4a	Opus	3.7
D4	MP3, M4a	Ogg	3.82

results significantly. Furthermore, we introduced a new method of online augmentation - SpecAverage, that contributed to our results. The combination of our methods achieved state of the art results in the DF category, both with a single system and with a system fusion scheme. In addition, the use of our methods achieved very strong performance on the ASVspoof 2019 evaluation data set as well. Given that our methods are mostly used on audio frequency based features, we believe that they can be used in other audio related tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by Israeli Innovation Authority as part of Nofar program, the German Research Foundation (DFG) via the German-Israeli Project Cooperation [DIP], and by the ISF research grant 899/21.

References

Besacier, Laurent, Mayorga, P, Bonastre, J-F, Fredouille, Corinne, Meignier, Sylvain, 2003. Overview of compression and packet loss effects in speech biometrics. IEE Proc. Vis. Image Signal Process. 150 (6), 372–376.

Bosi, Marina, Brandenburg, Karlheinz, Quackenbush, Schuyler, Fielder, Louis, Akagiri, Kenzo, Fuchs, Hendrik, Dietz, Martin, 1997. ISO/IEC MPEG-2 advanced audio coding. J. Audio Eng. Soc. 45 (10), 789–814.

Chen, Tianxiang, Khoury, Elie, Phatak, Kedar, Sivaraman, Ganesh, 2021a. Pindrop labs' submission to the asvspoof 2021 challenge. In: Proc. ASVspoof 2021 Workshop.

Chen, Tianxiang, Kumar, Avrosh, Nagarsheth, Parav, Sivaraman, Ganesh, Khoury, Elie, 2020. Generalization of audio deepfake detection.In: Proc. Odyssey 2020 the Speaker and Language Recognition Workshop.pp. 132–137.

Chen, Xinhui, Zhang, You, Zhu, Ge, Duan, Zhiyao, 2021b. UR channel-robust synthetic speech detection system for ASVspoof 2021. arXiv preprint arXiv:2107.12018.

Cheuk, Kin Wai, Agres, Kat, Herremans, Dorien, 2020. The impact of audio input representations on neural network based music transcription. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–6.

Das, Rohan Kumar, Yang, Jichen, Li, Haizhou, 2021. Data augmentation with signal companding for detection of logical access attacks. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6349–6353.

Delgado, Héctor, Evans, Nicholas, Kinnunen, Tomi, Lee, Kong Aik, Liu, Xuechen, Nautsch, Andreas, Patino, Jose, Sahidullah, Md, Todisco, Massimiliano, Wang, Xin, et al., 2021. ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan. arXiv preprint arXiv:2109.00535.

Dou, Yongqiang, Yang, Haocheng, Yang, Maolin, Xu, Yanyan, Ke, Dengfeng, 2021. Dynamically mitigating data discrepancy with balanced focal loss for replay attack detection. pp. 4115–4122.

Dutoit, Thierry, 1997. An Introduction to Text-to-Speech Synthesism, Vol. 3. Springer Science & Business Media.

Ferras, Marc, Madikeri, Srikanth, Motlicek, Petr, Dey, Subhadeep, Bourlard, Hervé, 2016. A large-scale open-source acoustic simulator for speaker recognition. IEEE Signal Process. Lett. 23 (4), 527–531.

Hailu, Nirayo, Siegert, Ingo, Nürnberger, Andreas, 2020. Improving automatic speech recognition utilizing audio-codecs for data augmentation. In: 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP). IEEE, pp. 1–5.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2016. Deep residual learning for image recognition. pp. 770–778.

Hua, Guang, Bengjinteoh, Andrew, Zhang, Haijian, 2021. Towards end-to-end synthetic speech detection. IEEE Signal Process. Lett..

Jarina, Roman, Polackỳ, Jozef, Počta, Peter, Chmulík, Michal, 2017. Automatic speaker verification on narrowband and wideband lossy coded clean speech. IET Biometr. 6 (4), 276–281.

Kinnunen, Tomi, Delgado, Héctor, Evans, Nicholas, Lee, Kong Aik, Vestman, Ville, Nautsch, Andreas, Todisco, Massimiliano, Wang, Xin, Sahidullah, Md, Yamagishi, Junichi, et al., 2020. Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals. IEEE/ACM Trans. Audio, Speech Lang. Process. 28, 2195–2210.

Kinnunen, Tomi, Sahidullah, Md, Delgado, Héctor, Todisco, Massimiliano, Evans, Nicholas, Yamagishi, Junichi, Lee, Kong Aik, 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection.

Kobayashi, Kazuhiro, Huang, Wen-Chin, Wu, Yi-Chiao, Tobing, Patrick Lumban, Hayashi, Tomoki, Toda, Tomoki, 2021. crank: An open-source software for nonparallel voice conversion based on vector-quantized variational autoencoder. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5934–5938.

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

Lavrentyeva, Galina, Novoselov, Sergey, Tseren, Andzhukaev, Volkova, Marina, Gorlanov, Artem, Kozlov, Alexandr, 2019. STC antispoofing systems for the ASVspoof2019 challenge. arXiv preprint arXiv:1904.05576.

Li, Xu, Li, Na, Weng, Chao, Liu, Xunying, Su, Dan, Yu, Dong, Meng, Helen, 2021a. Replay and synthetic speech detection with res2net architecture. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6354–6358.

Li, Xu, Wu, Xixin, Lu, Hui, Liu, Xunying, Meng, Helen, 2021b. Channel-wise gated res2net: Towards robust detection of synthetic speech attacks. arXiv preprint arXiv:2107.08803.

Loshchilov, Ilya, Hutter, Frank, 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Luo, Anwei, Li, Enlei, Liu, Yongliang, Kang, Xiangui, Wang, Z Jane, 2021. A capsule network based approach for detection of audio spoofing attacks. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6359–6363.

Luo, Wenjie, Li, Yujia, Urtasun, Raquel, Zemel, Richard, 2016. Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 4905–4913.

Mayorga, Pedro, Besacier, Laurent, Lamy, Richard, Serignat, J-F, 2003. Audio packet loss over IP and speech recognition. In: 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721). IEEE, pp. 607–612.

Mermelstein, Paul, 1988. G. 722: a new CCITT coding standard for digital transmission of wideband audio signals. IEEE Commun. Mag. 26 (1), 8–15.

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

Paul, Dipjyoti, Pal, Monisankha, Saha, Goutam, 2017. Spectral features for synthetic speech detection. IEEE J. Sel. Top. Sign. Proces. 11 (4), 605–617.

Robert, James, Webbie, Marc, et al., 2018. Pydub.

Sriskandaraja, Kaavya, Sethu, Vidhyasaharan, Le, Phu Ngoc, Ambikairajah, Eliathamby, 2016. Investigation of sub-band discriminative information between spoofed and genuine speech. In: Interspeech. pp. 1710–1714.

Stauffer, A.R., Lawson, Aaron D., 2009. Speaker Recognition on Lossy Compressed Speech Using the Speex Codec. Tech. Rep., Research Associates for Defense Conversion (RADC) Marcy NY.

Sterne, Jonathan, 2012. MP3: The Meaning of a Format. Duke University Press.

Tak, Hemlata, Jung, Jee-weon, Patino, Jose, Todisco, Massimiliano, Evans, Nicholas, 2021. Graph attention networks for Anti-Spoofing. arXiv preprint arXiv:2104.03654.

Tak, Hemlata, Patino, Jose, Nautsch, Andreas, Evans, Nicholas, Todisco, Massimiliano, 2020a. An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification. arXiv preprint arXiv:2004.06422.

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

Todisco, Massimiliano, Wang, Xin, Vestman, Ville, Sahidullah, Md, Delgado, Héctor, Nautsch, Andreas, Yamagishi, Junichi, Evans, Nicholas, Kinnunen, Tomi, Lee, Kong Aik, 2019. ASVspoof 2019: Future horizons in spoofed and fake audio detection. arXiv preprint arXiv:1904.05441.

Tomar, Suramya, 2006. Converting video formats with FFmpeg. Linux J. 2006 (146), 10.

Vu, Thi-Ly, Zeng, Zhiping, Xu, Haihua, Chng, Eng-Siong, 2019. Audio codec simulation based data augmentation for telephony speech recognition. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, pp. 198–203.

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

Wu, Zhizheng, Kinnunen, Tomi, Evans, Nicholas, Yamagishi, Junichi, Hanilçi, Cemal, Sahidullah, Md, Sizov, Aleksandr, 2015. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In: Sixteenth Annual Conference of the International Speech Communication Association.

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

Yang, Jichen, Wang, Hongji, Das, Rohan Kumar, Qian, Yanmin, 2021. Modified magnitude-phase spectrum information for spoofing detection. IEEE/ACM Trans. Audio, Speech Lang. Process. 29, 1065–1078.

Zhang, You, Jiang, Fei, Duan, Zhiyao, 2021a. One-class learning towards synthetic voice spoofing detection. IEEE Signal Process. Lett. 28, 937–941;
Zhang, You, Jiang, Fei, Duan, Zhiyao, 2021b. One-class learning towards synthetic voice spoofing detection. IEEE Signal Process. PP, 1.

Zhang, You, Zhu, Ge, Jiang, Fei, Duan, Zhiyao, 2021c. An empirical study on channel effects for synthetic voice spoofing countermeasure systems. arXiv preprint arXiv:2104.01320.

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