Low-quality Fake Audio Detection through Frequency Feature Masking

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

Sunmook Choi
Korea University
Seoul, Republic of Korea
felixchoi@korea.ac.kr

Jonghoon Yang
Chung-Ang University
Seoul, Republic of Korea
yjhoon2@cau.ac.kr

Yerin Lee
Chung-Ang University
Seoul, Republic of Korea
dldpfls14@cau.ac.kr

Soyul Han
Chung-Ang University
Seoul, Republic of Korea
soyul5458@cau.ac.kr

Seungsang Oh
Korea University
Seoul, Republic of Korea
seungsang@korea.ac.kr

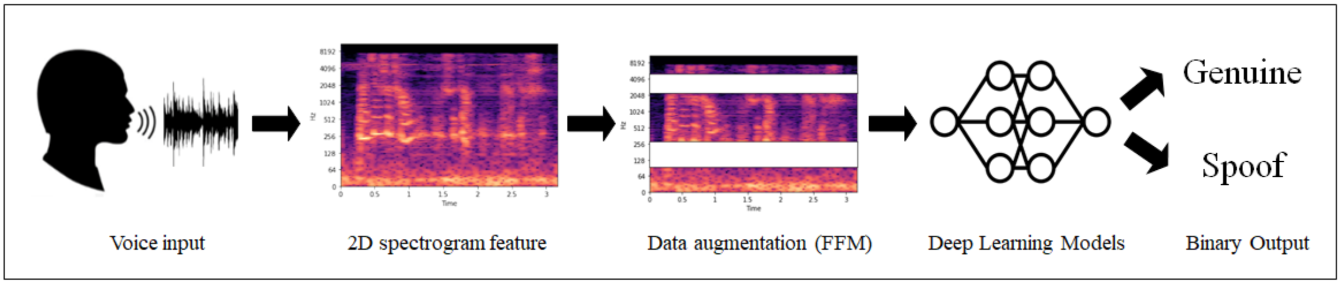


Figure 1: Overview of the proposed system.

ABSTRACT

The first Audio Deep Synthesis Detection Challenge (ADD 2022) competition was held which dealt with audio deepfake detection, audio deep synthesis, audio fake game, and adversarial attacks. Our team participated in track 1, classifying bona fide and fake utterances in noisy environments. Through exploratory data analysis, we found that noisy signals appear in similar frequency bands for given voice samples. If a model is trained to rely heavily on information in frequency bands where noise exists, performance will be poor. In this paper, we propose a data augmentation method, Frequency Feature Masking (FFM) that randomly masks frequency bands. FFM makes a model robust by not relying on specific frequency bands and prevents overfitting. We applied FFM and mixup augmentation on five spectrogram-based deep neural network architectures that performed well for spoofing detection using mel-spectrogram and constant Q transform (CQT) features. Our best submission achieved 23.8% in EER and ranked 3rd on track 1. To demonstrate the usefulness of our proposed FFM augmentation, we further experimented with FFM augmentation using ASVspoof 2019 Logical Access (LA) datasets.

DDAM '22, October 14, 2022, Lisboa, Portugal.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9496-3/22/10…$15.00
https://doi.org/10.1145/3552466.3556533

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Supervised learning by classification*; *Batch learning*.

KEYWORDS

Audio deep synthesis, low-quality audio, deep learning, frequency feature masking, audio data augmentation

ACM Reference Format:
Il-Youp Kwak, Sunmook Choi, Jonghoon Yang, Yerin Lee, Soyul Han, and Seungsang Oh. 2022. Low-quality Fake Audio Detection through Frequency Feature Masking. In *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia (DDAM '22), October 14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3552466.3556533

1 INTRODUCTION

Voice-related technologies are rapidly developing in recent years, prompting several high-tech companies to develop voice assistants. Voice assistants are being made to include various functions based on the company's business. For example, a manufacturing firm, Samsung Electronics wants to use Bixby to control their devices, and an e-commerce company, Amazon lets Alexa purchase products.

However, the more helpful features were introduced, the more security issues arose. Burger King, for example, advertised in 2017: "OK Google, what is the Whopper burger?" and, Google Home was activated immediately and read Wikipedia's description of Whopper burger. We do not want a user's voice assistant to follow the instructions of a television or for others to control the voice assistant using the user's synthesized voices.

Spoofing detection competitions have been held steadily to address these issues. These competitions include AVspoof 2015 [5], ASVspoof 2015 [28], ASVspoof 2017 [11], ASVspoof 2019 [22], ASV spoof 2021 [29], and ADD 2022 [31]. Many results have been published for spoofing detection. Light CNN (LCNN) using max feature maps (MFM) as an activation of a convolution block showed good performance over 2017, 2019, and 2021 years [14, 15, 23, 26]. Attempts to create deep learning model's based on raw audio rather than a spectrogram signal basis have been investigated [21, 25], and AASIST model was proposed based on graph attention network [7].

ADD 2022 was the most recent competition, and Track 1 addressed spoofing detection in noisy environments. The data with environmental noise was a new factor which was not considered in previous competitions. We found that environmental noise signals such as refrigerator sound reside in a similar frequency band for given voice samples. We propose a frequency feature masking (FFM) augmentation technique that masks a high-frequency region, a low-frequency region, or a random frequency band where environmental noises are expected to exist. FFM helps robust training of a spoofing detection model in noisy environments. FFM is similar to SpecAugment [18] in that the augmentation policy comprises wrapping features, masking blocks of frequency channels, or time steps. FFM is designed to focus more on voice spoofing problems in noisy environments. Fig. 1 describes an overview of our proposed system. Our method converts speech input data into spectrogram features, and mixup [33] and FFM augmentation are applied to the features. Later, we trained five spectrogram-based deep neural network architectures that performed well for spoofing detection using mel-spectrogram and constant Q transform (CQT) features. Five systems are (1) ResMax [13] with CQT feature, (2) Light CNN (LCNN) [15, 26] with CQT feature, (3) BC-ResMax [30] (a variant of BC-ResNet [8]) with mel-spectrogram feature, (4) Double Depthwise Separable net (DDWSnet) [30] with mel-spectrogram feature, and (5) Overlapped Frequency-Distributed (OFD) model [2] with CQT feature. The ensemble model of those five systems was submitted in the competition.

Although the effectiveness of the FFM augmentation method was demonstrated in ADD 2022 competition, further research is necessary to establish what advantages FFM offers in broader settings. In this study, we test the FFM augmentation technique by using ADD competition dataset, which includes noisy scenarios as well as ASVspoof 2019 LA data in general situations. The performance of the models are measured by the equal error rate (EER). The following is a summary of our works.

(1) We describe our submitted ensemble system comprising LCNN, ResMax, BC-ResMax, DDWS, and OFD models. It achieved 23.8% in EER, ranking 3rd on track 1 in ADD 2022 competition.

(2) Using ADD competition dataset, the application of FFM produced exceptional results. Without any data augmentations, BC-ResMax, DDWS, LCNN, ResMax, and OFD models have EER of 22.31%, 22.53%, 20.14%, 19.97%, and 21.35%, respectively. Mixup augmentation decreased those EERs to 15.87%, 17.19%, 20.13%, 15.60%, and 17.12%. Additional FFM augmentation remarkably decreased EER of BC-ResMax, DDWS, LCNN, and ResMax models to 12.09%, 13.40%, 16.81%, and 15.22%.

(3) FFM was useful not only on the noisy ADD dataset but also on the ASVspoof 2019 LA dataset. In particular, the high-frequency masking significantly improved the performance on the ASVspoof 2019 LA dataset. The performance of baseline models without any augmentation ranged from 2.63% to 4.45% in EER. Additional mixup augmentation achieved EER ranging from 2.87% to 4.06%. On the other hand, LCNN with mixup, Low-frequency, and High-frequency masking augmentation achieved the best result with 1.93% in EER.

2 METHODS

2.1 Feature engineering

We utilized CQT and mel-spectrogram feature extractions using the librosa software [16]. For the CQT feature extraction, we set the minimum frequency to 5, the number of frequency bins to 100, and the filter scale factor to 1. For the mel-spectrogram feature extraction, we used 100 frequency bins, 1024 window lengths, and 512 hop sizes.

2.2 Mixup augmentation

Mixup is a widely used data augmentation in voice classifications as well as image classifications [33]. The method mixes two different samples of the training set according to a parameter λ which is sampled from the $beta(\alpha, \alpha)$ distribution with a hyper parameter α. In our modeling, we set α from 0.4-0.9. The method is as follows:

$$X = \lambda X_i + (1 - \lambda) X_j, \tag{1}$$

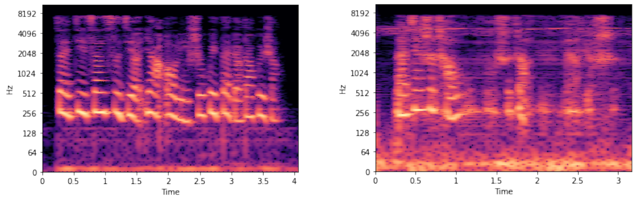
$$y = \lambda y_i + (1 - \lambda) y_j, \tag{2}$$

where X_i and X_j are different spectrogram images with their corresponding labels y_i and y_j, respectively.

2.3 Frequency feature masking (FFM) augmentation

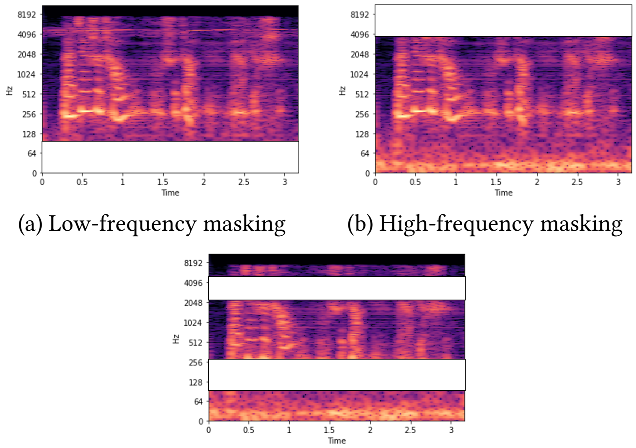
Fig. 2(a) and (b) show a genuine voice sample in a normal environment from training data and in a noisy environment from adaptation data, respectively. In Fig. 2(b), we can find long horizontal lines around 4096 Hz. The utterance sample had a fixed high frequency noise signal like the sound of a vacuum cleaner when we heard the utterance sample. We can think of the following scenarios:

- High frequency areas may have more noise. Let the model focus more on other frequency areas by masking high frequency areas.
- Low frequency areas may have more noise. Let the model focus more on other frequency areas by masking low frequency areas.
- If a model is trained to focus on specific frequency bands, then the model performance will decrease when a noise signal occurs in the frequency band. Random frequency band masking helps the model learn without being biased in some frequency band.



(a) Normal environment (b) Noisy environment

Figure 2: A genuine voice sample in a normal environment from training data (a), and in a noisy environment from adaptation data (b).



(a) Low-frequency masking (b) High-frequency masking

(c) Random frequency band masking

Figure 3: Description on Low-frequency masking, High-frequency masking and Random frequency band masking.

To address these scenarios, we propose a ‘frequency feature masking (FFM)‘ augmentation, which considers three types of masking, ‘low-frequency masking’, ‘high-frequency masking’, and ‘random frequency band masking’ (shown in Fig. 3). FFM augmentation is applied for each training sample, and the details for each technique are explained as follows.

Low-frequency masking is applied with probability p_l. The low-frequency area is randomly selected (ex. from 0 to 14 range in 100 mel bins), and all corresponding values in the spectrogram are set to 0.

High-frequency masking is similar to low-frequency masking. Only differences are the masking probability p_h and the selected frequency area (ex. from 85 to 100 range in 100 mel bins).

Random frequency band masking is applied with probability p_r. Also the number of masked frequency bands, band sizes, and the locations of the bands are randomly selected. The values in the masked frequency bands are set to 0.

2.4 Models

Using the previously proposed FFM augmentation, five spectrogram-based models were considered.

2.4.1 *LCNN model.*

LCNN model has proven useful in ASVspoof 2017, 2019, and 2021 competitions [14, 15, 23]. We used a deeper LCNN model by adding more layers to Light CNN-9 model, which repeats five convolution layers and four network-in-network (NIN) layers [26]. Our deeper model iterates six convolution layers and five NIN layers using 32, 48, 64, 32, 32, and 32 convolution filters and 32, 48, 64, 64, and 32 NIN filters. As in the previous model, the kernel size of the first convolution layer is set to 5, and those of the remaining convolution layers are set to 3. Fig. 4(a) describes the base LCNN block that consists of a convolution, MFM and an optional batch normalization layer (dotted block applied when $b = 1$). Fig. 5(a) describes full model architecture. LCNN blocks with kernel size 1 are NIN layers. Except for the second convolution layer, batch normalization follows. All NIN layers are followed by batch normalization layer. Instead of using a fully connected layer defined in Light CNN-9 model [26], we used a global average pooling layer, batch normalization and a dropout layer with probability 0.5.

2.4.2 *ResMax model.*

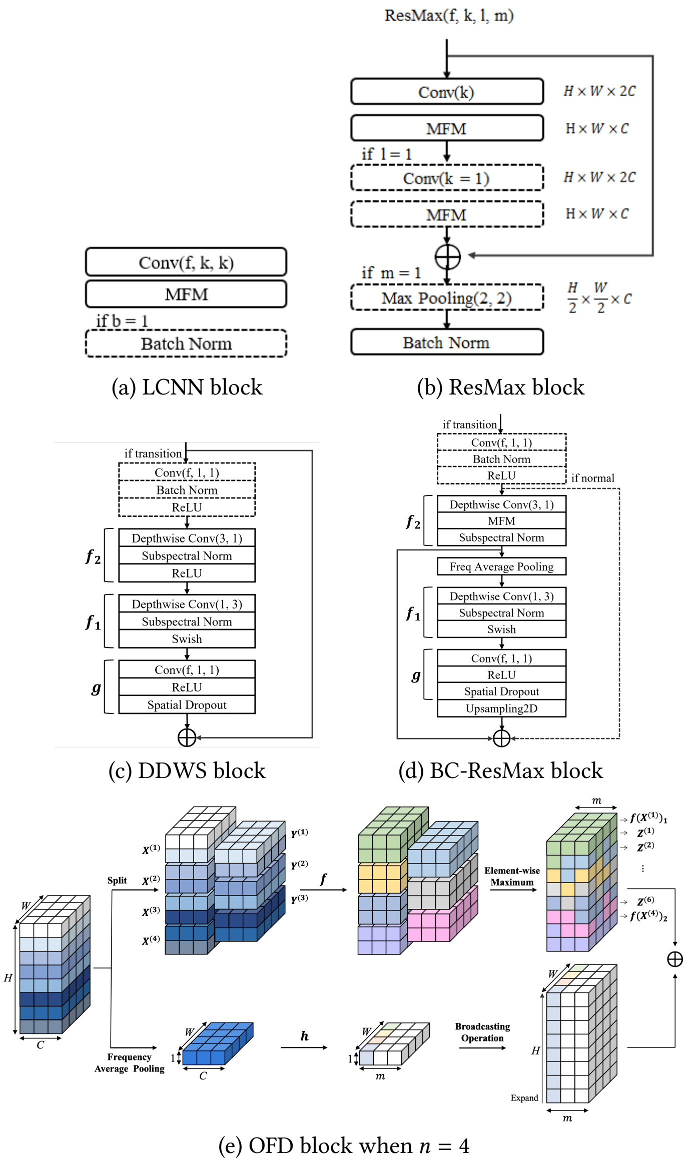
We previously proposed ResMax model and confirmed that it showed excellent performance in the ASVspoof 2019 competition dataset [13]. A ResMax block has four parameters (f, k, l, m) as described in Fig. 4(b), indicating the number of ResMax filters (f), the kernel size in the convolution layer (k), whether the additional execution (described with the dotted line) in the ResMax block is performed (l), and whether a max-pooling is applied (m), respectively. An MFM layer is used after a convolution layer which takes maximum of two feature maps elementwise. This process makes the model robust and makes the model lighter through selection. ResMax blocks use a skip connection to mitigate performance degradation issues due to information loss. One architecture of the entire model is described in Fig. 5(b).

2.4.3 *Double Depthwise Separable (DDWS) model.*

The original depthwise separable convolution consists of a depthwise convolution with (k_1, k_2)-kernel and a pointwise convolution [3, 6]. In DDWS model, Yang et al. (2022) [30] proposed to apply the existing depthwise convolutions twice followed by a pointwise convolution. Two depthwise convolutions have the kernels of the size $(k_1, 1)$ and $(1, k_2)$, respectively, to consider frequency and temporal information separately. Precisely, we define a DDWS block as shown in Fig. 4(c). We define f_1 to be a depthwise convolution with $(1, k_2)$ kernel followed by Subspectral Normalization (SSN) [1] and Swish activation [19]. f_2 is a depthwise convolution with $(k_1, 1)$ kernel followed by SSN and ReLU activation [17]. Lastly, g is a composite of a pointwise convolution, ReLU activation, and spatial dropout. A block design without dotted marks in Fig. 4(c) on the top describes a normal block, which should be applied when an input and the output of the block have the same number of channels. A design with dotted marks is a transition block, which is applied when the number of channels becomes different after an input passes the block. The full model architecture is described in Fig. 5(c).

2.4.4 *BC-ResMax model.*

Yang et al. (2022) [30] revised BC-ResNet [8, 9] by integrating max feature map (MFM) activation from LCNN. Our BC-ResMax block is described in Fig. 4(d). We define f_2 to be a depthwise convolution with $(k_1, 1)$ kernel followed by MFM and SSN. As explained in

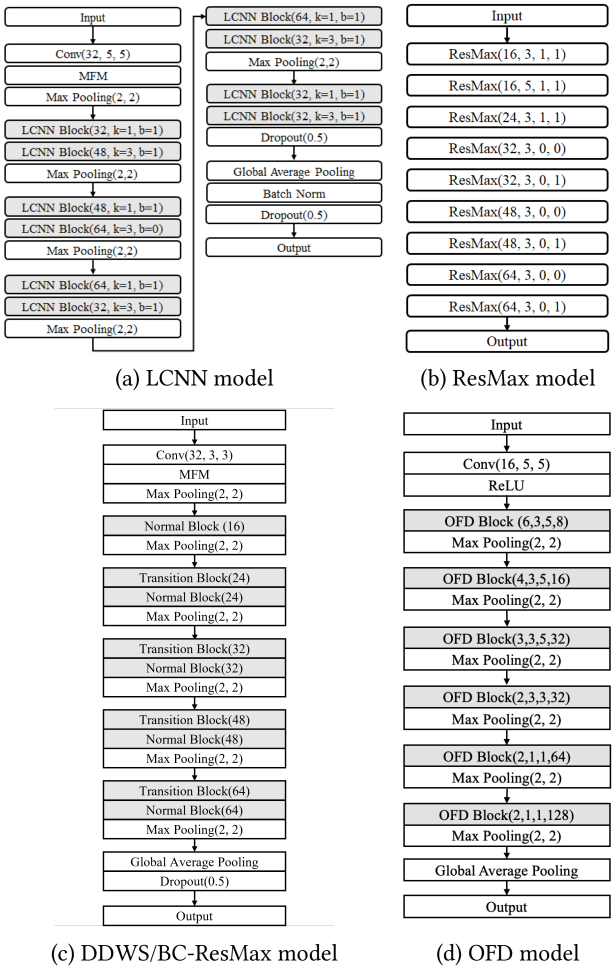


(a) LCNN block (b) ResMax block

(c) DDWS block (d) BC-ResMax block

(e) OFD block when $n = 4$

Figure 4: Model Blocks



(a) LCNN model (b) ResMax model

(c) DDWS/BC-ResMax model (d) OFD model

Figure 5: Model Architectures

DDWS model, a block design without dotted marks on the top is a normal block, and a design with dotted marks is a transition block. The full model architecture is described in Fig. 5(c).

2.4.5 *Overlapped Frequency-Distributed (OFD) model.*

Choi et al. (2022) [2] introduce Overlapped Frequency-Distributed (OFD) model. The model consists of six OFD blocks, each of which has two streams described in Fig. 4(e). The first stream is motivated by FreqCNN model [27] and the second stream is obtained from BC-ResNet model [8, 9].

The first stream is to learn frequency-related features. Let $X = (x_{ij}) \in \mathbf{R}^{H \times W}$ be an input of the block, where H and W correspond to the frequency and temporal dimensions, respectively, and the channel dimension is excluded. Let n be the split number along the frequency axis. We zero-pad the input X along the frequency axis so that H is a multiple of $2n$, if necessary, and set $s = \frac{H}{2n}$. We split X into n disjoint parts $\{X^{(k)}\}_{k=1}^{n}$ and additional $n-1$ overlapped parts $\{Y^{(k)}\}_{k=1}^{n-1}$, which are as follows:

$$X^{(k)} = (x_{ij}) \in \mathbf{R}^{2s \times W}, \quad 1 + 2(k-1)s \le i \le 2ks \tag{3}$$

$$Y^{(k)} = (x_{ij}) \in \mathbf{R}^{2s \times W}, \quad 1 + (2k-1)s \le i \le (2k+1)s \tag{4}$$

We'd like to mention that the lower half part of $X^{(k)}$ and the upper half part of $Y^{(k)}$ coincide, and the upper half part of $X^{(k+1)}$ and the lower half part of $Y^{(k)}$ coincide. We define f to be a composite of a convolution with $(k_1, 1)$-kernel, batch normalization, ReLU, another convolution with $(k_1, 1)$-kernel, and batch normalization. Notice that no activation is used in the second convolution, and zero-padding is required in each convolution to keep the size the same. Next, $f(X^{(k)})$'s and $f(Y^{(k)})$'s should be joined to reconstruct a feature map of the size $H \times W$. Each of them is divided into two parts, which are the upper and lower half parts, saying $f(X^{(k)})_1$, $f(X^{(k)})_2$, $f(Y^{(k)})_1$, and $f(Y^{(k)})_2$, having the size $s \times W$. Then, we define $Z^{(2k-1)} = f(X^{(k)})_2 \vee f(Y^{(k)})_1$ and $Z^{(2k)} = f(X^{(k+1)})_1 \vee$

$f(Y^{(k)})_2$ where $\vee$ is an elementwise maximum operation. Note that the maximum operation is indeed an MFM activation, acting as an activation in the last convolution. Finally, for the set of $2n$ sub-images

$$\left\{f(X^{(1)})_1,\ Z^{(1)},\ \ldots,\ Z^{(2n-2)},\ f(X^{(n)})_2\right\}, \tag{5}$$

we take concatenation of these images along the frequency axis. Then the resulting output is the final output of this stream.

The second stream is to learn temporal features. The input X is averaged out along the frequency axis so that we get a feature map of size $\mathbf{R}^{1\times W}$. The resulting feature map is put into the function h, where h is defined to be a composite of $(1, k_2)$-depthwise convolution with dilation of 4, batch normalization, swish activation, $(1, 1)$-convolution with ReLU activation, and spatial dropout. Here we use zero-padding to keep the temporal dimension the same. Lastly, we expand the feature map along the frequency axis using broadcasting operation [8].

Each OFD block is represented as 'OFD Block(n, k_1, k_2, m)' in the figure. Here, m is the number of filters used in convolutions in both streams. The overall architecture of OFD model is illustrated in Fig. 5(d). The dropout rate is fixed to 0.5, and notice that in the 5th and 6th blocks of OFD model, $(1, 1)$-kernel is used in all convolution layers ($k_1 = k_2 = 1$) to manage the receptive field in the network [12].

3 EXPERIMENTS

3.1 Datasets

3.1.1 *ADD 2022 challenge datasets.*

ADD 2022 challenge [31] used more realistically generated data to prevent spoofing attacks in real situations. This challenge comprised a low-quality fake audio detection (LF) track, a partially fake audio detection (PF) track, and an audio fake game (FG) track. We concentrate on track 1, which consists of genuine data and spoof-attack data generated by text-to-speech (TTS) and voice conversion (VC). The LF track classifies genuine and spoofing utterances interrupted by various real-world noises and background music. Table 1 describes the ADD challenge dataset. It consists of training, development (dev.), adaptation, and test data. Training and development sets use publicly available Mandarin AISHELL-3 [20] to select utterances and the number of these sets are about 27K, and 23K utterance samples. The adaptation data has about 1K utterances as in a similar environment to the test set. The test data consists of approximately 110K unlabeled utterances containing various noises.

Table 1: The number of utterances in training, development, adaptation and test sets of the ADD database.

	Training	Dev.	Adaptation	Test
Genuine	3,012	2,307	300	-
Fake	24,072	21,295	700	-
Unlabeled	-	-	-	109,199

3.1.2 *ASVspoof 2019 challenge LA dataset.*

ASVspoof 2019 is the challenge focused on TTS, VC and replay spoofing attack types. The ASVspoof 2019 dataset [22] consists of logical access (LA) and physical access (PA) scenarios. Both are derived from VCTK basic corpus [24], and the dataset is divided into three parts: training, development, and evaluation. We focuse on LA data since it focuses on TTS and VC attack types similar to the ADD challenge dataset. The LA scenario uses a variety of 17 state-of-the-art TTS and VC systems to generate bona fide and spoofed speech. Six TTS and VC systems are designated as known attacks in the training and development data, while the remaining eleven systems are designated as unknown attacks in the test data. Table 2 describes the ASVspoof 2019 LA dataset. It consists of training, development (dev.), and test data. Training and development data consist of about 25K and 25K utterances. The test data consists of approximately 71K utterances containing unknown attacks.

Table 2: The number of utterances in training, development and test sets of the ASVspoof 2019 LA database

	Training	Dev.	Test
Genuine	2,580	2,548	7,355
Fake	22,800	22,296	63,882

3.2 Experimental setup

We studied and compared our experimental results with the introduced models in the previous section. To verify the performance, FFM was applied to two databases: ADD 2022 competition dataset with noise scenarios and ASVspoof 2019 LA dataset in general situations. Each utterance is sampled at 16kHz. The hyperparameters in FFM augmentations are set as follows. For high-frequency (HF) masking, a random integer h is selected from 80 to 87, and the high-frequency band which ranges from h to 100 among 100 mel bins is randomly masked (with $p_h = 0.5$). For low-frequency (LF) masking, a random integer l is selected from 7 to 12, and the low-frequency band which ranges from 1 to l among 100 mel bins is randomly masked (with $p_l = 0.5$). For random frequency (RF) band masking, it is performed randomly with $p_r = 2/3$. Two bands are used for masking with probability $1/3$ and one band is used with probability $1/3$. The location(s) and window size(s) (between 8 to 12) of masking band(s) are randomly selected.

The performance is measured in equal error rate (EER) which is the rate where the false acceptance rate (FAR) and the false rejection rate (FRR) are equal. In general, the lower the EER value is, the better the model performs. We used a 9-second sample. For training the proposed models, we set the batch size to 16 and the epoch to 70. We ran three training sessions and obtained the average EER value for development and test data.The learning rate started from 1e-3 and decreased to 1e-5 using a learning rate scheduler with a sigmoidal decay function.

In the experiment using ADD data, training and development data were combined to learn the model to evaluate the adaptation test set. As a model for submitting ADD final evaluation, all training, development, and adaptation data were combined to train the final models. In the experiment using ASVspoof2019 data, training was conducted using only training data.

3.3 Experimental result on ADD 2022 dataset

We experimented with our proposed models and augmentation techniques on the adaptation set which exposed to real-world noises and background music. Table 3 describes EERs on the adaptation set for each model with different augmentation methods applied. Performance has improved more when applying the augmentation techniques rather than comparing the models. Without any data augmentations, BC-ResMax, DDWS, LCNN, ResMax, and OFD models have an EER of 22.31%, 22.53%, 20.14%, 19.97%, and 21.35%, respectively. Mixup augmentation reduced those EERs to 15.87%, 17.19%, 20.13%, 15.60%, and 17.12%, respectively. Moreover, additional FFM augmentation remarkably lowered to 12.09%, 13.40%, 16.81%, 15.22%, and 16.04% in EER, respectively. These results show that the proposed FFM augmentation considers the noisy environment effectively.

Table 3: EER (%) on the adaptation set for each model with different augmentation methods. The best results for each model are shown in bold.

Augmentation	BC-ResMax	DDWS	LCNN	ResMax	OFD
Baseline	22.31	22.53	20.14	19.97	21.35
Mixup	15.87	17.19	20.13	15.60	17.12
Mixup, LF	16.29	16.88	19.88	16.49	17.20
Mixup, HF	16.05	17.31	19.70	17.37	17.39
Mixup, RF	14.02	**13.40**	18.45	**15.22**	17.11
Mixup, LF, HF	14.90	15.99	19.56	19.62	19.60
Mixup, LF, RF	**12.09**	15.52	**16.81**	16.89	16.66
Mixup, HF, RF	12.64	15.60	20.24	17.33	**16.04**
Mixup, LF, HF, RF	12.77	16.70	19.00	17.48	17.78

3.4 Experimental result on ASVspoof 2019 LA dataset

Table 4 shows the results of applying our approach in the LA of the ASVspoof 2019 database. We experimented with eight different combinations. Without any data augmentations, BC-ResMax, DDWS, LCNN, ResMax, and OFD models have an EER of 3.83%, 2.63%, 3.18%, 4.45%, and 2.68%, respectively. Augmentation using Mixup, LF and HF showed the best performance in BC-ResMax, DDWS and LCNN with 2.47%, 2.45%, and 1.93%. On the other hand, ResMax and OFD models performed better with Mixup and HF augmentation resulting 2.08% and 1.49%, respectively, in EER. The performance with Mixup, LF and HF augmentation was excellent in the ASVspoof LA dataset regardless of the models. Especially, HF augmentation was highly effective. When only Mixup was applied, the EERs ranged from 2.87% to 4.06%. Whereas when HF was additionally applied with Mixup, there was a huge improvement in EERs, ranging from 1.49% to 2.73%. We may infer two conclusions: the high-frequency area of LA data might slightly harm the performance in discrimination, or the test samples have different patterns in the high-frequency part from the training samples.

Table 4: EER (%) of different droping out augmentation methods on ASVspoof 2019 LA dataset. The best results for each model are shown in bold.

Augmentation	BC-ResMax	DDWS	LCNN	ResMax	OFD
Baseline	3.83	2.63	3.18	4.45	2.68
Mixup	3.67	4.06	3.11	3.51	2.87
Mixup, LF	3.20	3.28	2.88	3.60	2.58
Mixup, HF	2.73	2.66	2.04	**2.08**	**1.49**
Mixup, RF	5.17	6.65	3.52	4.57	3.25
Mixup, LF, HF	**2.47**	**2.45**	**1.93**	2.15	1.88
Mixup, LF, RF	4.99	4.47	3.30	4.97	3.32
Mixup, HF, RF	4.06	4.78	2.56	3.21	1.99
Mixup, LF, HF, RF	4.34	4.33	2.45	3.78	2.49

3.5 Submitted ensemble system

Table 5 describes five top-performing single systems, data augmentation methods applied, their EER on final evaluation data, weights for the final ensemble model, and the EER of our ensemble system. Applying all LF, HF and RF augmentations in the final evaluation data did not always produce the best result, and we have tested various combinations and hyperparameters. The final models were selected to have as few correlations as possible, and the ensemble weights were heuristically chosen based on EER.

Table 5: EER (%) on the final evaluation data from the ADD 2022 challenge, and weights for ensemble model.

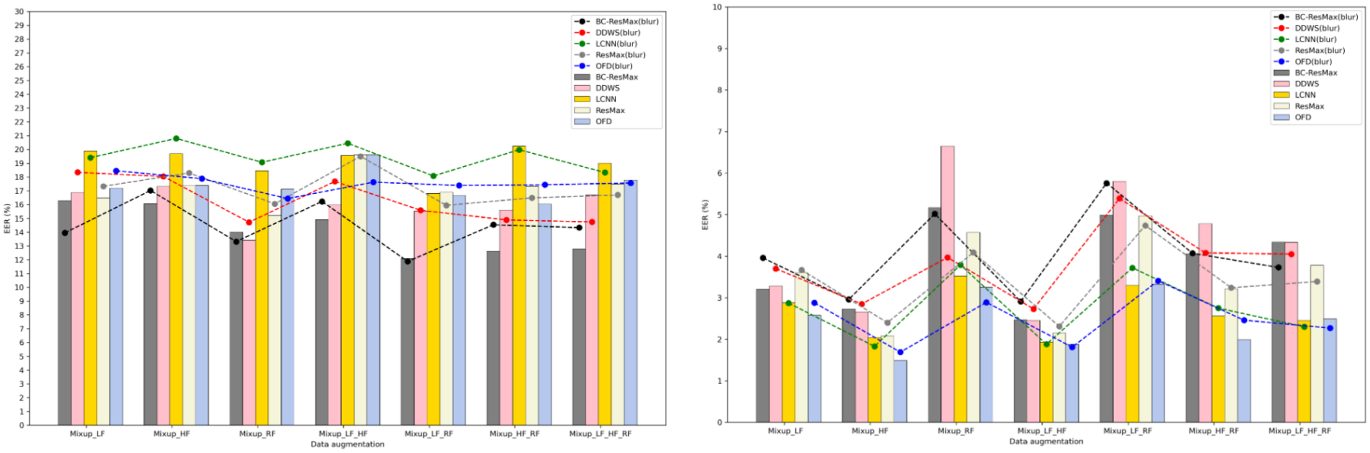
Model	Feature	Augmentation	EER	weights
LCNN	CQT	Mixup, LF, RF	26.05%	0.20
ResMax	CQT	Mixup, RF	24.7%	0.27
DDWS	melspec	Mixup, RF	26.40%	0.20
BC-ResMax	melspec	Mixup, LF, RF	27.34%	0.13
OFD	CQT	Mixup	26.02%	0.20
Ensemble	-	-	23.8%	-

3.6 Ablation Study

3.6.1 *Masking vs Blurring.*

Tomilov et al. [23] pointed out that using finite impulse response (FIR) filters improves the performance of spoofing detection models. Using a high pass filter or a low pass filter of FIR is similar to the HF or LF augmentation of FFM, respectively. Main difference is that if we use a filter of FIR, the filter does not mask the information of a frequency band fully but it weakens the information. We thought of modifying our proposed FFM method to play a role in weakening information in a frequency band. We conducted an ablation study to verify the effectiveness of *Masking* and *Blurring* in the proposed FFM augmentation. Masking is the original proposed method which drops the selected frequency band to '0' as shown in Fig. 3. Blurring is a method to diminish information by multiplying the selected frequency band by a tiny number 0.01 instead of deleting the selected frequency band.

Fig. 6 compares Masking and Blurring methods applied on ADD 2022 dataset and ASV2019 LA dataset. Bar graphs represent the EERs of models when Masking methods are applied, and dotted line graphs represent the EERs when Blurring methods are applied. Table 6 and 7 show the quantitative results of applying Blurring to the ADD 2022 and ASVspoof 2019 LA datasets. The best EER values for BC-ResMax, DDWS, LCNN, ResMax, and OFD models applied



(a) ADD 2022 database

(b) ASVspoof 2019 LA database

Figure 6: EER (%) of droping out mask and Blurring mask in (a) ADD 2022 dataset and in (b) ASVspoof 2019 LA dataset.

with FFM augmentation on ADD 2022 dataset are 11.88%, 14.72%, 17.98%, 15.95%, and 16.44% respectively. BC-ResMax model gave the best EER using Blurring, resulting an EER of 11.88%, which is a better result than using Masking. As a result of applying Blurring with DDWS, LCNN, and ResMax, the overall EER value was about 1%p higher than the performance of applying Masking. The best EER values for BC-ResMax, DDWS, LCNN, ResMax, and OFD models in the ASVspoof 2019 dataset with Blurring FFM augmentation applied are 2.91%, 2.73%, 1.83%, 2.31%, and 1.69%, respectively. Compared to Masking, LCNN model gave even better result with Blurring application method, with an EER of 1.83%. On the other hand, the other models showed better performance with Masking. Overall, the difference between Blurring augmentation and Masking augmentation was small, and there was no statistically significant difference using pairwise t-test. We should try more simulations to obtain any statistical implication. the HF and LF augmentation of FFM is similar to the method using FIR filters that reduce information in high and low frequency bands (high pass and low pass filters). FIR filters are applied on raw audio to reduce high- or low-frequency information [23]. This can be interpreted as blurring high- or low-frequency bands from the spectrogram point of view. Instead of FFM masking, an experiment was conducted on the blurring method, but no significant difference was found in our ablation study.

Table 6: EER (%) of different Blurring augmentation methods on ADD 2022 dataset.

Augmentation	BC-ResMax	DDWS	LCNN	ResMax	OFD
Mixup, LF	13.95	18.34	19.41	17.33	18.45
Mixup, HF	17.02	18.04	20.80	18.29	17.88
Mixup, RF	13.32	**14.72**	19.07	16.06	**16.44**
Mixup, LF, HF	16.23	17.68	20.45	19.50	17.63
Mixup, LF, RF	**11.88**	15.59	**18.08**	**15.95**	17.39
Mixup, HF, RF	14.54	14.89	19.98	16.49	17.44
Mixup, LF, HF, RF	14.33	14.74	18.33	16.70	17.57

Table 7: EER (%) of different Blurring augmentation methods on ASVspoof 2019 LA dataset.

Augmentation	BC-ResMax	DDWS	LCNN	ResMax	OFD
Mixup, LF	3.96	3.70	2.87	3.80	2.88
Mixup, HF	2.96	2.85	**1.83**	2.40	**1.69**
Mixup, RF	5.02	3.97	3.79	4.09	2.89
Mixup, LF, HF	**2.91**	**2.73**	1.88	**2.31**	1.81
Mixup, LF, RF	5.76	5.39	3.72	4.74	3.41
Mixup, HF, RF	4.07	4.08	2.75	3.24	2.46
Mixup, LF, HF, RF	3.73	4.05	2.30	3.39	2.27

4 DISCUSSION

4.1 Comparison with FIR and SpecAugment

As seen in the previous ablation study, the HF and LF augmentation of FFM is similar to the method using FIR filters that reduce information in high and low frequency bands (high pass and low pass filters). FIR filters are applied on raw audio to reduce high- or low-frequency information [23]. This can be interpreted as blurring high- or low-frequency bands from the spectrogram point of view. Instead of FFM masking, an experiment was conducted on the blurring method, but no significant difference was found in our ablation study.

The SpecAugument [18] consists of time warping, frequency masking, and time masking. FFM has a modified version of policy considering only frequency masking without time warping and time masking. A slightly different part of the policy part is that SpecAugment specifies how many frequency bands to mask as a hyperparameter. On the other hand, in our proposed FFM, one among 0, 1, 2, 3, ..., K is randomly selected for frequency band masking, and K is designated as a hyperparameter. Comparison of the differences between the two policies should be further studied.

4.2 When to use FFM

FFM is an augmentation technique that considers noisy test environments. Efficacy was proven through experimental results using ADD data, and LF and HF showed good performance even on general ASVspoof2019 data, not in noisy environments. However, there was no performance improvement in RF. In the ASVspoof 2019 problem, the frequency band in which the human voice exists will be more important. By randomly masking low or high frequencies that contain a small amount of human voice, the importance of the relevant part is reflected less, and the model performance may have been improved. In a general test situation like the ASVspoof 2019 data, it will be necessary to test which FFM method is better.

4.3 Future works

There are many data augmentation techniques used in the voice field [4, 10, 18, 32]. In this study, an augmentation technique considering a noisy environment was proposed and used together with mixup [33], but comparison with other augmentation techniques is insufficient. Therefore, other masking-based augmentation methods and additional comparative experiments are needed.

5 CONCLUSION

This paper covers the models used by our team participating in ADD 2022 Track 1. Unlike other voice spoofing detection contests, ADD 2022 Track 1 considers spoofing detection in a situation exposed to real-life noise. We proposed FFM augmentation to robustly train a model against real-life noise, which is a spectrogram-based augmentation technique. We designed five spectrogram-based spoofing detection models; LCNN with CQT feature, ResMax with CQT feature, DDWS with mel-spectrogram feature, BC-ResMax with mel-spectrogram feature, and OFD with CQT feature. The final ensemble model achieved 23.8% EER, placing 3rd in the ADD 2022 Track 1 competition. The best baseline given in the competition for track 1 [31] showed a performance of 24.1% in EER, and there were only three teams that outperformed the baseline among the 42 participating teams.

To verify the usefulness of FFM augmentation, experiments were conducted on ADD 2022 and ASVspoof 2019 LA datasets. It was confirmed that FFM improved the performance in all experimental models. Interestingly, FFM was useful not only on the noisy ADD 2022 dataset but also on ASVspoof 2019 LA dataset. In particular, the HF augmentation significantly improved the performance on ASVspoof 2019 LA dataset. The high-frequency part of the ASVspoof 2019 LA test dataset might differ from the high-frequency part of the training set when compared with other frequency bands, or high-frequency parts were less critical for classification models. For indirect comparison with the FIR filter method, Blurring augmentation method was also tested, and there was no statistically significant difference between the blurring augmentation and masking augmentation. It would be because of small number of simulation runs, and we should increase the number of runs in the future study.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. NRF-2022R1F1A1064273 and NRF-2020R1C1C1A01013020).

REFERENCES

[1] Simyung Chang, Hyoungwoo Park, Janghoon Cho, Hyunsin Park, Sungrack Yun, and Kyuwoong Hwang. 2021. Subspectral Normalization for Neural Audio Data Processing. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Toronto, 850–854.

[2] Sunmook Choi, Il-Youp Kwak, and Seungsang Oh. 2022. Overlapped Frequency-Distributed Network: Frequency-Aware Voice Spoofing Countermeasure. In *Proc. Interspeech 2022*. ISCA, Incheon.

[3] François Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Honolulu, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

[4] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. , 8 pages.

[5] Serife Kucur Ergunay, Elie Khoury, Alexandros Lazaridis, and Sebastien Marcel. 2015. On the vulnerability of speaker verification to realistic voice spoofing. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, Piscataway, 1–6. https://doi.org/10.1109/BTAS.2015.7358783

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* abs/1704.04861 (2017), 1–9. arXiv:1704.04861

[7] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. 2022. AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Brno, 6367–6371. https://doi.org/10.1109/ICASSP43922.2022.9747766

[8] Byeonggeun Kim, Simyung Chang, Jinkyu Lee, and Dooyong Sung. 2021. Broadcasted Residual Learning for Efficient Keyword Spotting. In *Proc. Interspeech 2021*. ISCA, Brno, 4538–4542.

[9] Byeonggeun Kim, Seunghan Yang, Jangho Kim, and Simyung Chang. 2021. *QTI Submission to DCASE 2021: Residual Normalization for Device-Imbalanced Acoustic Scene Classification with Efficient Design.* Technical Report. DCASE2021 Challenge.

[10] Gwantae Kim, David K. Han, and Hanseok Ko. 2021. SpecMix : A Mixed Sample Data Augmentation Method for Training with Time-Frequency Domain Features. In *Proc. Interspeech 2021*. ISCA, Brno, 546–550. https://doi.org/10.21437/Interspeech.2021-103

[11] Tomi Kinnunen, Md Sahidullah, Hector Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In *Proc. Interspeech 2017*. ISCA, Stockholm, 2–6.

[12] Khaled Koutini, Hamid Eghbal-zadeh, and Gerhard Widmer. 2021. Receptive Field Regularization Techniques for Audio Classification and Tagging With Deep Convolutional Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 1987–2000. https://doi.org/10.1109/TASLP.2021.3082307

[13] Il-Youp Kwak, Sungsu Kwag, Junhee Lee, Jun Ho Huh, Choong-Hoon Lee, Youngbae Jeon, Jeonghwan Hwang, and Ji Won Yoon. 2021. ResMax: Detecting Voice Spoofing Attacks with Residual Network and Max Feature Map. In *25th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, Milan, 4837–4844.

[14] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. 2017. Audio Replay Attack Detection with Deep Learning Frameworks. In *Proc. Interspeech 2017*. ISCA, Stockholm, 82–86.

[15] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov. 2019. STC Antispoofing Systems for the ASVspoof2019 Challenge. In *Proc. Interspeech 2019*. ISCA, Graz, 1033–1037. https://doi.org/10.21437/Interspeech.2019-1768

[16] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, Vol. 8. SciPy, Austin, 1–7.

[17] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Haifa, Israel) *(ICML'10)*. Omnipress, Madison, WI, USA, 807–814.

[18] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*. ISCA, Graz, 2613–2617.

[19] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941* 7, 1 (2017), 5.

[20] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. , 5 pages. arXiv:2010.11567

[21] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. 2021. End-to-End anti-spoofing with RawNet2. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Brno, 6369–6373. https://doi.org/10.1109/ICASSP39728.2021.9414234

[22] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi H. Kinnunen, and Kong Aik Lee. 2019. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *Proc. Interspeech 2019*. ISCA, Graz, 1008–1012. https://doi.org/10.21437/Interspeech.2019-2249

[23] Anton Tomilov, Aleksei Svishchev, Marina Volkova, Artem Chirkovskiy, Alexander Kondratev, and Galina Lavrentyeva. 2021. STC Antispoofing Systems for the ASVspoof2021 Challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. ISCA, Brno, 61–67. https://doi.org/10.21437/ASVSPOOF.2021-10

[24] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. 2017. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. https://datashare.ed.ac.uk/handle/10283/2651

[25] Jee weon Jung, Seung bin Kim, Hye jin Shim, Ju ho Kim, and Ha-Jin Yu. 2020. Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms. In *Proc. Interspeech 2020*. ISCA, Online, 1496–1500. https://doi.org/10.21437/Interspeech.2020-1011

[26] X. Wu, R. He, Z. Sun, and T. Tan. 2018. A Light CNN for Deep Face Representation With Noisy Labels. *IEEE Transactions on Information Forensics and Security* 13, 11 (Nov 2018), 2884–2896. https://doi.org/10.1109/TIFS.2018.2833032

[27] Yu Wu, Hua Mao, and Zhang Yi. 2018. Audio classification using attention-augmented convolutional neural network. *Knowledge-Based Systems* 161 (2018), 90–100.

[28] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilci, Md. Sahidullah, and Aleksandr Sizov. 2015. ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge. In *Proc. Interspeech 2015*. ISCA, Dresden, 2037–2041.

[29] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. 2021. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. ISCA, Brno, 47–54. https://doi.org/10.21437/ASVSPOOF.2021-8

[30] Jonghoon Yang, Sunmook Choi, Yerin Lee, Seungsang Oh, and Il-Youp Kwak. 2022. Light-weight Frequency Information Aware Neural Network Architecture for Voice Spoofing Detection. In *26th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, Montreal Quebec.

[31] Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhan Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, and Haizhou Li. 2022. ADD 2022: the First Audio Deep Synthesis Detection Challenge. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, Singapore, 9216–9220.

[32] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Seoul, 6023–6032.

[33] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. , 13 pages.