



Novel Augmentation Schemes for Device Robust Acoustic Scene Classification

Sukanya Sonowal¹, Anish Tamse²

¹Samsung Electronics

²FasterAI

suk.sonowal@gmail.com, anish@fasterai.io

Abstract

For audio classification tasks, one has access to the recordings from only a few microphones while the system could be deployed for a wider range of microphones. This paper discusses augmentation methods for audio scene recognition with the aim of improving performance on recordings from unseen microphones. The proposed augmentation schemes can be broadly classified into two categories. The first category, which is called the frequency response augmentation technique, aims to artificially generate ‘new’ microphone frequency responses. This is achieved by collecting microphone impulse responses from a publicly available library and applying image augmentation techniques on them to create a more diverse set of frequency responses. The train data is then augmented with these artificially generated frequency responses. The second category consists of the amplitude augmentation and random frame drop methods which are simple yet effective in further boosting the performance. We test all these augmentation methods on various architectures and observe a good classification accuracy of 76.0% on the DCASE 2020 Task 1a set. Especially on unseen devices our best reported accuracy, without using any model ensembles, is 74.24%.

Index Terms: audio scene classification, data augmentation, microphone impulse response, device robustness, transformers, convolutional networks

1. Introduction

Acoustic scene classification (ASC) is the task of identifying the environment (scene) in which the sound was recorded. Acoustic scenes comprise a diverse range of sound events and can sometimes even share similar sounds, making the task challenging [1]. Additionally, there is also the problem of device robustness, posed by the variation in characteristics of recording devices [2].

Proposed methods to solve these challenges include new or improved feature extraction techniques [3], [4], [5], [6], [7], data augmentation strategies [1], [2], [8], deep feature learning methods [9], [10], network architectures [11], [12], [13], [14], loss functions [15] and other schemes [16], [17]. Although these methods have shown good classification performance, we wonder if there can be a more targeted approach to solve the problem of device robustness, which remains a challenging area of research. It is often difficult to collect sufficient recordings from many different devices/microphones and it is likely that in a real production environment, an ASC system will be exposed to recordings from unseen devices. Even for a single microphone, its frequency response characteristics might be dependent on the orientation and other mechanical factors [18], [19]. To mitigate this, [15] proposes the use of spectrum correction and heated-up softmax, [20] proposes a channel conversion technique using factorized hierarchical variational autoen-

coder and [1] uses CliqueNets and a mixture-of-experts (MoEs) layer. [2] proposes a wide variety of augmentation schemes which include mixup, random crop, SpecAugment, pitch shift, speed change and the addition of noise and reverberation. They also mention that some of these augmentation methods can help simulate ‘new’ devices and report improved accuracies on the DCASE 2020 Task 1a set [21], which is a multi-device dataset.

In this paper we propose two new types of augmentation methods: (i) The first type is aimed at simulating ‘new’ devices and we call this the frequency response (FR) augmentation technique. As we can artificially synthesize recordings of a device (to a good extent) from its frequency response, our approach is to synthetically create many such frequency responses. These responses are then used to augment the train data, with the hope that the augmented data will be representative of recordings from an unseen realistic microphone. For this, we gather microphone impulse responses (IR) from the freely available Microphone Impulse Response Project (MicIRP) library [22]. This library contains around 65 IRs of vintage microphones from 30 manufacturers [23]. Interestingly, when these IRs are used as-is to augment the train data of the DCASE set, we already see performance improvements. In order to further increase the data diversity, we then propose methods to artificially create new frequency responses by applying different transformations to the frequency responses corresponding to these IRs. When these artificially generated frequency responses are used to augment the data, we observe a much higher performance improvement, especially on devices that were not present in the train data. (ii) Under the next type, we introduce two general purpose augmentation methods, which are, namely, amplitude augmentation and random frame drop. These seemingly simple methods bring about further performance gains on top of the earlier performance improvements.

All the proposed augmentation methods are tested extensively on the Resnet and the Convolutional vision Transformer (CvT) [24] model, which was recently introduced for image classification. We use Resnet for benchmarking as it is the most widely used architecture for ASC. To demonstrate the model agnostic behavior of our augmentation techniques we choose CvT, as its Transformer based architecture is quite different from the Resnet. To the best of our knowledge, we are the first to employ CvT for ASC tasks. Our proposed methods show improvements on both these models and our best reported accuracy of 76.0% on the DCASE 2020 Task 1a development set is higher than that of the winning submission for DCASE 2020 Task 1a challenge, which is 74.4%. On unseen devices, our best reported accuracy of 74.24% is also higher than that of the second best submission, which is 73.0% (winning submission did not report their unseen device performance).

2. Proposed methods

2.1. Frequency response augmentation

The motivation behind this augmentation is to create a diverse pool of frequency responses, to account for the wide range of microphones available in the real world. The MicIRP library contains IRs of 65 different microphones which are available in wav format. We take the N-point Fast Fourier Transform (FFT) of each of these IRs to get a pool of 65 frequency responses F . In the training stage, a frequency response f_i , which is a 1D array of size N , is selected by sampling uniformly from F . One of the following transformations is then randomly selected and applied to it.

2.1.1. Stretch

As the name suggests, this augmentation linearly stretches f_i in the frequency domain. The output f_i^{st} of this transformation can be expressed as $f_i^{st}(x) = f_i(x/s_t)$, where $s_t > 1$. This transformation is implemented using the `torchvision.transforms.functional.resize` function, which basically resizes the input to a desired size by taking f_i and the desired length as inputs. The desired length in this case is $N \cdot s_t$. As s_t is greater than 1, the length of the output is greater than N . In this case, we only use the values of the output from 0 to $N - 1$. The rest of the values are discarded. s_t is randomly sampled from a uniform distribution $U(1, 1.3)$.

2.1.2. Squeeze

This is the opposite of stretch augmentation, with $s_t < 1$. The same `torchvision.transforms.functional.resize` function is used to implement this. The only difference is that zero-padding is performed to increase the output length to N . As before, s_t is randomly sampled from a uniform distribution $U(0.6, 1)$.

2.1.3. Squeeze with low frequency bound

This is similar to the above squeeze transformation. The only difference is that the values of f_i corresponding to frequencies in the range $[0, f_{sq.thr}]$ Hz are not modified. The rest of the array f_i which corresponds to frequencies in range $(f_{sq.thr}, F_s/2]$ Hz, where F_s is the sampling frequency, is modified using the technique mentioned above. In our implementation $f_{sq.thr}$ is set to 1000 Hz.

2.1.4. Shift

This transformation introduces the effect of right-shift in the frequency domain on f_i by a shift parameter s_h , which is sampled from a uniform distribution $U(0, 5000)$ Hz. Additionally, we use the concept of low frequency bound ($f_{sh.thr}$) similar to above i.e.

$$f_i^{sh}(x) = \begin{cases} f_i(x) & \text{if } x \leq f_{sh.thr} + s_h \\ f_i(x - s_h) & \text{if } x > f_{sh.thr} + s_h \end{cases} \quad (1)$$

where f_i^{sh} is the output of the transformation and the hyperparameter $f_{sh.thr}$ is set to 3000 Hz.

All of the above transformations are visualized in Figure 1. After applying one of these transformations randomly on f_i , the output f_i^{aug} is then multiplied with the short-time Fourier transform (STFT) X of the input audio signal as shown

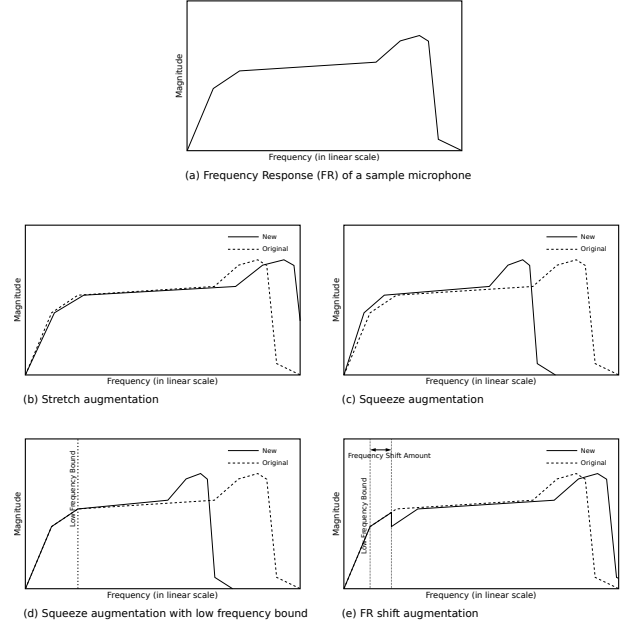


Figure 1: Visualizations for the four frequency response augmentation schemes described in Section 2.1. (a) shows a sample FR used as the source FR for visualizing the four augmentation schemes. (b) shows the stretch augmentation, wherein each frequency from the original FR maps to a higher frequency in the augmented FR. (c) shows the squeeze augmentation where each frequency from the original FR maps to a lower frequency in the augmented FR. (d) visualizes the squeeze augmentation with low frequency bound, which is similar to the previous augmentation except that below a certain frequency threshold, the augmented FR matches the original FR. (e) visualizes the shift augmentation, where after a threshold frequency the augmented FR is the right shifted variant of the original FR.

below

$$X_m(x, t) = X(x, t) f_i^{aug}(x) \quad (2)$$

where X_m is the augmented STFT and x and t are the frequency and time indices respectively. Ideally, deconvolution should be first performed on the input signal to remove the effects of the microphone it was recorded on. However, since straightforward deconvolution can be tricky and can produce artifacts in the signal [25], for the sake of simplicity we approximate this process by using the input audio samples from the best microphone available.

2.2. General augmentation methods

2.2.1. Amplitude augmentation

There can be multiple sound events of different amplitude levels within an audio clip. The amplitude levels can vary due to factors like distance between the audio source and microphone, loudness of the source of the sound etc. The motivation behind this augmentation is to make the system more robust towards the variations in amplitude levels. For this, we introduce random perturbations in the amplitude values of the input STFT features by multiplying it with the values of $m(t)$, which is a sinusoidal function of time t . The equation below describes $m(t)$

$$m(t) = 1 + a \cdot \sin(bt) \quad (3)$$

Table 1: *Probability of selection of each augmentation. For the FR augmentations, one of the five options are selected based on the probabilities mentioned below. This is followed by amplitude augmentation. Finally random frame drop is always applied.*

Augmentation type	Selection probability
Stretch	19%
Squeeze	19%
Squeeze w/ low bound	25%
Shift	17%
No FR augmentation	20%
Amplitude augmentation	30%
No Amplitude augmentation	70%
Random frame drop	100%

where a and b correspond to the amplitude and frequency values respectively of the sinusoid. In our implementation, a is sampled from $U(0, 0.02)$ and b is chosen such that there are between 2.5 to 10.5 sinusoidal cycles. The augmented STFT is obtained as $X(x, t)m(t)$.

2.2.2. Random frame drop

In this simple method, we randomly remove one or more strips of contiguous time frames from the input features. This is similar to the time masking concept of the SpecAugment [26] method. The difference is that instead of assigning some predefined values (or zeros) to the features within the time strip, in this method we remove those strips of frames entirely. The total number of frames to be dropped from an input feature is not fixed and is randomly chosen from $U(1, 12)$ for a mini batch. After this, random crop with crop length 400 is performed on the remaining input features [12].

2.3. Network architectures

The augmentation methods are tested separately on the Resnet and CvT models. The structure of the Resnet model is similar to that of [12]. In this structure, there are two separate pathways for the low and high frequencies that are fused towards the end and there is also no frequency sub-sampling in the network. The network has approximately 5.6M parameters. CvT is a Transformer based model. Transformer models have recently gained wide popularity in solving tasks related to Natural Language Processing (NLP) and computer vision. These models consist of stacked self attention layers which help capture both local and global dependencies. The Convolutional vision Transformer (CvT) model was recently proposed in [24] with an aim to combine both Convolutional Neural Networks (CNNs) and Transformers for image classification. The model was then shown to perform better than both Resnet and other Transformer-based models, while utilizing fewer parameters. Based on these observations, we use the CvT model for our ASC task. In the paper, the authors introduce three CvT architectures which differ in the number of Transformer blocks and hidden feature dimensions. For our task, we use the CvT-13 model which is the most lightweight among the three. This model has approximately 20M trainable parameters. Rather than training the model from scratch, we fine-tune the ImageNet-pretrained CvT-13 model (which is made available by the authors) for ASC. For this, we only replace the last fully

connected layer of the model in accordance with the number of audio scene labels.

The rationale behind the choice of these two architectures is to have two models sufficiently different from each other. Resnet being the commonly used network for ASC is the most obvious choice. CvT, unlike Resnet, combines convolutions with attention style architecture.

3. Experiments

The experiment results are evaluated on the DCASE 2020 Task 1a development data set. This is a multi-device data set consisting of around 17000 24 bit audio clips sampled at 44.1 KHz. Around 3000 audio clips of this set are test samples. The training samples come from six different devices, three of which are real microphones (A, B and C) and the rest are simulated (s1, s2 and s3). The test data contains audio samples from three additional simulated microphones (s4, s5 and s6) which are not present in the training set.

For each input sample, STFT with 2048 FFT points is computed using a window size of 2048 samples and a hop length of 1024 samples. The STFT computation is performed using the Librosa library [27]. FR augmentation and amplitude augmentation are applied to the result of STFT at this stage. Also, they are randomly performed with probabilities as shown in Table 1, which are determined from experiments. The log-mel filter bank (LMFB) features are extracted along with log-mel deltas and delta-deltas from the STFT features using the TorchAudio library [28]. The result is of the size $423 \times 256 \times 3$ which is fed to the networks. The random frame drop method is always applied to these LMFB features. In case of FR augmentation, we multiply the frequency responses with the STFT of audios from device A only. This is because among devices A, B and C, device A is closest to an ideal mic with a flat response.

Each result presented is the average of three runs for the given configuration. For each run, the training is performed for 191 epochs unless specified otherwise. Stochastic gradient descent (SGD) with a cosine-decay-restart learning rate scheduler is used to train all the networks as done in [2], [29].

Each result presents the average result on seen (A, B, C, s1, s2, s3) and unseen devices (s4, s5, s6) along with the overall average. Unseen devices are those whose samples are absent in the training set. The presentation of results this way is done to emphasize the effectiveness of augmentation techniques on the cross device accuracy.

As shown in Table 2, without any data augmentation, the Resnet model achieves a classification accuracy of 67.5% while the CvT model achieves 68.6% classification accuracy. Applying the commonly used augmentation techniques of mixup and random crop improves the result for both the models. For Resnet the performance increases to 72.4% and for CvT, it increases to 70.1%. We consider this as the base performance. The impact of each proposed augmentation technique is described next.

3.1. FR augmentation

Firstly, a frequency response is randomly selected from the pool F and is directly multiplied with the STFT of the input audio. There is no transformation applied to the selected frequency response. Noticeably, this alone increases the performance of both the models. For the Resnet, the classification accuracy on unseen microphones increases from 68.8% to 71.5% and for the CvT it increases from 62.1% to 67.9%. The next four rows of

Table 2: *Experimental results*

Item	System	Accuracy for device (%)		
		seen	unseen	overall
1	Baseline [30]	58.9	44.3	54.2
2	Suh et al. [29]	-	-	74.4
3	Hu et al. (Resnet) [2]	76.0	71.0	74.6
4	Hu et al. (FCNN) [2]	78.9	73.0	76.9
<hr/>				
Resnet				
1	no crop or mixup	69.0	64.4	67.5
2	crop + mixup	74.2	68.8	72.4
3	(2) + 65 mic FR	74.3	71.5	73.4
4	(3) + stretch	74.7	71.4	73.6
5	(3) + squeeze	74.8	72.4	74.0
6	(3) + squeeze w/ freq bound	74.7	72.1	73.8
7	(3) + shift	74.7	72.2	73.9
8	(4) + (5) + (6) + (7)	74.9	72.8	74.2
9	(8) + amplitude	75.1	72.0	74.0
10	(9) + drop	74.4	73.1	74.0
11	(9) + SpecAugment	75.1	72.7	74.3
12	(10) + SpecAugment	75.4	73.7	74.8
<hr/>				
CvT				
1	no crop or mixup	71.3	63.3	68.6
2	crop + mixup	74.1	62.1	70.1
3	(2) + 65 mic FR	75.1	67.9	72.7
4	(3) + stretch	75.0	70.3	73.4
5	(3) + squeeze	75.5	69.8	73.6
6	(3) + squeeze w/ freq bound	75.9	69.5	73.7
7	(3) + shift	76.5	69.5	74.1
8	(4) + (5) + (6) + (7)	76.1	71.9	74.7
9	(8) + amplitude	77.1	73.6	76.0
10	(9) + drop	77.0	74.0	76.0
11	(9) + SpecAugment	76.6	74.2	75.8
12	(10) + SpecAugment	76.9	73.5	75.8

the results show the individual impact of application of each of the transformations described in Section 2.1. The results show that the individual transformations can additionally boost the performance especially for unseen devices for both the Resnet and CvT models. Finally, we combine all the four transformations and achieve an overall accuracy of 74.2% with Resnet and 74.7% with the CvT model. Hence the use of FR augmentation boosts the overall accuracy for the Resnet by 1.8% and the accuracy on unseen devices by 4.0%. The overall accuracy improvement for CvT is 4.6% and the improvement on unseen microphones is 9.8%. The performance of seen and unseen devices are both significantly higher than the base performance.

3.2. General augmentation

Items 9 and 10 in Table 2 show the result of using the amplitude augmentation and random frame drop methods. These augmentations are applied after applying the FR augmentation to the input STFT. Additionally item 11 shows the result of replacing random frame drop with SpecAugment. This is done to compare random frame drop with SpecAugment. Item 12 shows the result of combining SpecAugment with all the proposed methods. The general augmentations do not improve the classification performance on Resnet significantly. The correspond-

ing improvement on CvT however is quite significant, which is 1.3%. The discrepancy in the improvement on Resnet and CvT is likely due to the model complexity. CvT having a lot more parameters can effectively make use of the extra augmented data to generalize well. This effect was also visible during FR augmentation where the improvement on CvT was higher than that on Resnet. Of the two general augmentation schemes applied, most of the improvement is obtained from the amplitude augmentation scheme. In the case of the CvT model, it improves the unseen device accuracy by 1.7%. The random frame drop method improves the unseen device accuracy, however it lowers the seen device accuracy. The final best performing model is the CvT model combined with all the proposed augmentation enabled during the training. The classification accuracy achieved on unseen devices is 74.0% with the overall accuracy being 76.0%.

3.3. Comparison with competitive methods

Our results are compared with three systems. The first is the official baseline system [30] of the DCASE 2020 challenge. The second system [29] is the winning submission of the challenge. In this system, the authors propose the ‘Trident Resnet’ model in which the low frequency pathway is further split into two. The third system [2] is the second best submission of the challenge, in which the authors propose the use of several augmentation strategies and test them on Resnet and FCNN [31] models. We compare their non-ensemble Resnet and FCNN model results with ours for a fair comparison. It should be noted that even though the accuracy of the winning submission is less than that of the second best submission on this set, the winning system performs better on the final evaluation set. As both these systems use Resnet, we first compare their results with our Resnet results. Our best accuracy for the Resnet model is 74.8% and is higher than these two Resnet systems. In case of unseen devices, our best accuracy of 73.7% is 1.7% higher than that of the second best submission (winning submission did not report their device wise accuracy). Our best result for the CvT model is 76.0% which is higher than that of the winning submission but slightly lower than that of the FCNN model of the second best submission. However, if we compare the unseen device performance, our best accuracy of 74.2% is higher than that of their FCNN model by 1.2% which implies better model generalization.

4. Conclusions

Two key broad augmentation techniques are proposed, namely the FR augmentation technique and the amplitude augmentation technique. Significant performance improvement of around 6% is achieved by using the two together. Especially noticeable is the unseen device performance improvement of around 12%. This is due to the quality of augmented data generated using the proposed techniques. Also proposed is the random frame drop technique which improves the unseen device accuracy. The overall performance achieved exceeds the performance of the winning submission for DCASE 2020 Task 1a set by 1.6%. The augmentation techniques are not specific to the network either, as the performance improvement is demonstrated on two very different networks.

5. References

- [1] T. Nguyen and F. Pernkopf, ‘Acoustic scene classification with mismatched devices using cliquenets and mixup data augmenta-

- tion.” in *Interspeech*, 2019, pp. 2330–2334.
- [2] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, “A two-stage approach to device-robust acoustic scene classification,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 845–849.
 - [3] H. Chen, P. Zhang, and Y. Yan, “An audio scene classification framework with embedded filters and a dct-based temporal module,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 835–839.
 - [4] Y. Wu and T. Lee, “Enhancing sound texture in cnn-based acoustic scene classification,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 815–819.
 - [5] S. S. R. Phayeb, E. Benetos, and Y. Wang, “Subspectralnet – using sub-spectrogram based convolutional neural networks for acoustic scene classification,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 825–829.
 - [6] L. D. Pham, I. V. McLoughlin, H. Phan, and R. Palaniappan, “A robust framework for acoustic scene classification,” in *INTER-SPEECH*, 2019, pp. 3634–3638.
 - [7] Y. Liu, A. Neophytou, S. Sengupta, and E. Sommerlade, “Cross-modal spectrum transformation network for acoustic scene classification,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 830–834.
 - [8] S. Mun, S. Park, D. K. Han, and H. Ko, “Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane,” *Proc. DCASE*, pp. 93–97, 2017.
 - [9] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and A. Mertins, “Deep feature embedding and hierarchical classification for audio scene classification,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.
 - [10] V. Abrol and P. Sharma, “Learning hierarchy aware embedding from raw audio for acoustic scene classification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1964–1973, 2020.
 - [11] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, “Acoustic scene classification with squeeze-excitation residual networks,” *IEEE Access*, vol. 8, pp. 112 287–112 296, 2020.
 - [12] M. D. McDonnell and W. Gao, “Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 141–145.
 - [13] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, “The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification,” in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.
 - [14] L. Zhang, J. Han, and Z. Shi, “Learning temporal relations from semantic neighbors for acoustic scene classification,” *IEEE Signal Processing Letters*, vol. 27, pp. 950–954, 2020.
 - [15] T. Nguyen, F. Pernkopf, and M. Kosmider, “Acoustic scene classification for mismatched recording devices using heated-up softmax and spectrum correction,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 126–130.
 - [16] J. W. Jung, H. S. Heo, H. J. Shim, and H. J. Yu, “Knowledge distillation in acoustic scene classification,” *IEEE Access*, vol. 8, pp. 166 870–166 879, 2020.
 - [17] Y. Wu and T. Lee, “Time-frequency feature decomposition based on sound duration for acoustic scene classification,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 716–720.
 - [18] B. Series, “Effect of microphone directivity regarding level calibration and equalization of advanced sound systems,” 2018.
 - [19] J. G. Webster, *Mechanical Variables Measurement-Solid, Fluid, and Thermal*. CRC Press, 1999.
 - [20] S. Mun and S. Shon, “Domain mismatch robust acoustic scene classification using channel information conversion,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 845–849.
 - [21] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60. [Online]. Available: <https://arxiv.org/abs/2005.14623>
 - [22] “Microphone impulse response project.” [Online]. Available: <http://micirp.blogspot.com/>
 - [23] “Vintage mics irs - impulse responses - audiothing.” [Online]. Available: <https://www.audiothing.net/uncategorized/vintage-mics/>
 - [24] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “Cvt: Introducing convolutions to vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.
 - [25] A. Farina, “Advancements in impulse response measurements by sine sweeps,” in *Audio engineering society convention 122*. Audio Engineering Society, 2007.
 - [26] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.
 - [27] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.
 - [28] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhersch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, “Torchaudio: Building blocks for audio and speech processing,” *arXiv preprint arXiv:2110.15018*, 2021.
 - [29] S. Suh, S. Park, Y. Jeong, and T. Lee, “Designing acoustic scene classification models with cnn variants,” *Tech. Rep., DCASE2020 Challenge*, 2020.
 - [30] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions,” *arXiv preprint arXiv:2005.14623*, 2020.
 - [31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.