INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

SpecAugment++: A Hidden Space Data Augmentation Method for Acoustic Scene Classification

Helin Wang[1], *Yuexian Zou*[1,2,*], *Wenwu Wang*[3]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Center for Vision, Speech and Signal Processing, University of Surrey, UK
wanghl15@pku.edu.cn, zouyx@pku.edu.cn, w.wang@surrey.ac.uk

Abstract

In this paper, we present SpecAugment++, a novel data augmentation method for deep neural networks based acoustic scene classification (ASC). Different from other popular data augmentation methods such as SpecAugment and mixup that only work on the input space, SpecAugment++ is applied to both the input space and the hidden space of the deep neural networks to enhance the input and the intermediate feature representations. For an intermediate hidden state, the augmentation techniques consist of masking blocks of frequency channels and masking blocks of time frames, which improve generalization by enabling a model to attend not only to the most discriminative parts of the feature, but also the entire parts. Apart from using zeros for masking, we also examine two approaches for masking based on the use of other samples within the mini-batch, which helps introduce noises to the networks to make them more discriminative for classification. The experimental results on the DCASE 2018 Task1 dataset and DCASE 2019 Task1 dataset show that our proposed method can obtain 3.6% and 4.7% accuracy gains over a strong baseline without augmentation (*i.e.* CP-ResNet) respectively, and outperforms other previous data augmentation methods.

Index Terms: data augmentation, hidden space, feature masking, acoustic scene classification

1. Introduction

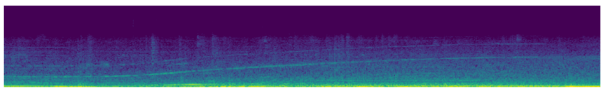
Deep learning has been successfully applied to various problems in Detection and Classification of Acoustic Scenes and Events (DCASE) [1–3], where convolutional neural networks (CNNs) [4–6], recurrent neural networks (RNNs) [7] and convolutional recurrent neural networks (CRNNs) [8] are often used as the network architectures. However, due to the lack of training data, these models are prone to overfitting [9].

Data augmentation methods have been widely used to overcome the problem of the limited data in the DCASE community, including the waveform-based and spectrogram-based data augmentation methods. Among the waveform-based methods, cropping is one of the common and effective approaches [10–12]. Salamon and Bello [13] proposed the usage of additional training data generated by time stretching, pitch shifting, dynamic range compression, and adding background noise chosen from an external dataset, which are also applied to the raw waveform. In [14], Park *et al.* proposed SpecAugment, an augmentation method that operates on the log mel spectrogram of the input audio and achieved the state-of-the-art performance for automatic speech recognition (ASR). SpecAugment consists of three kinds of deformations of the log mel spectrogram (*i.e.* time warping, time masking and frequency masking) to generate difficult-recognition samples, and converts ASR from an over-fitting to an under-fitting problem. However, these augmentation methods are only applied to the input of the deep neural networks, while augmenting the hidden space is not studied. Mixup [15, 16] and Between-Class (BC) learning [17] are also popular data augmentation methods for the DCASE tasks, which generate new data samples by mixing multiple audio samples and design the learning method for training a model to output the prediction of the mixed samples. SpeechMix [18] is a generalization of BC learning and it interpolates latent representations of hidden states. However, it is hard to determaine the label of the mixed audio as the energy and the distribution of the raw audio is quite different. In addition, Chen *et al.* [19] utilized the auxiliary classifier GAN (ACGAN) to generate fake samples for data augmentation, however, an extra discriminator is needed, as a result, the convergence of the networks becomes more difficult.

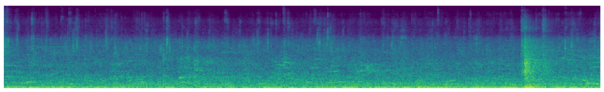
In this paper, we propose a novel data augmentation method (*i.e.* SpecAugment++), which is applied to both the input spectrograms and the hidden states of the deep neural networks. Inspired by the SpecAugment [14], two kinds of deformations of the hidden states (*e.g.* the intermediate feature maps in CNN) are employed, containing masking blocks of frequency channels and masking blocks of time frames. These methods improve the generalization ability of a model by allowing it to attend not only to the most discriminative time frames and frequency channels of audio, but also to the entire temporal frames and frequency channels. Time warping is not applied because it contributes little to the classification [14,20]. Apart from exploiting the augmentation for the hidden space, we study the influence of three schemes for time masking and frequency masking. The first is zero-value masking (ZM), which directly masks consecutive time frames and frequency channels with zeros. The other two schemes are the mini-batch based mixture masking (MM) and the mini-batch based cutting masking (CM), which utilize the time frames and frequency channels of other samples within the mini-batch for masking. These methods can be seen as introducing additional noises generated from the dataset, and guide the networks to be more discriminative for classification. Different from the mixup [15] and BC learning [17], the labels of the augmented data are not changed so that the training is more stable. The proposed method is simple and computationally cheap to apply, which is evaluated on two benchmark acoustic scene classification (ASC) datasets (DCASE 2018 [21] and DCASE 2019 [22] ASC task 1A datasets) and outperforms the state-of-the-art data augmentation methods.

*Corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-140

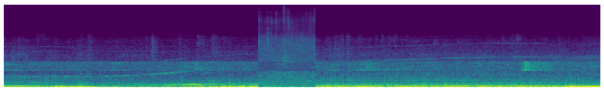


(a) The target sample.

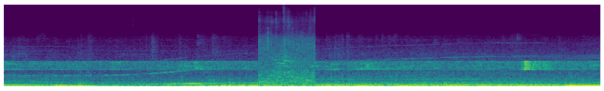


(b) Another sample within the same mini-batch.

(c) The augmented sample by ZM.



(d) The augmented sample by MM.



(e) The augmented sample by CM.

Figure 1: *Augmentations applied to the target sample. Here, the input log mel spectrogram is shown as an example, similarly to the augmentation applied to the intermediate hidden states. From top to bottom, the figures depict the log mel spectrogram of the target sample with no augmentation, another sample within the same mini-batch, the zero-value masking (ZM) applied, the mini-batch based mixture masking (MM) applied and the mini-batch based cutting masking (CM) applied.*

2. Proposed Method

In this section, we introduce the proposed augmentation method, which is constructed to act on either the input spectrograms or the intermediate hidden states of the neural networks. Time masking and frequency masking are employed, and the motivation is to promote the networks to be robust to deformations in the partial loss of the temporal information and the partial loss of the frequency information in each layer. Three types of masking schemes are developed, aiming to make better use of the data within the mini-batch. The augmentation techniques and the masking schemes are detailed as follows.

2.1. Augmentation Techniques

Let $x \in \mathbb{R}^{T \times F}$ denote the intermediate hidden state (or the input spectrogram), where T and F denote the number of time frames and frequency channels, respectively. Time masking is applied so that t consecutive time frames $[t_0, t_0 + t]$ are masked (which means replacing the elements by zeros or other values), where t is chosen from a uniform distribution from 0 to the time mask parameter t', and t_0 is chosen from $[0, T - t]$. Similarly, frequency masking is applied so that f consecutive frequency channels $[f_0, f_0 + f]$ are masked, where f is first chosen from a uniform distribution from 0 to the frequency mask parameter f', and f_0 is chosen from $[0, F - f]$. To simplify, the same time masking and frequency masking (*i.e.* the same t_0, t, f_0 and f) are applied to each intermediate hidden state in the same layer for a training sample. For instance, if there are C feature maps in the l-th layer of a CNN model, these C feature maps share the same temporal and frequency regions for masking of a single sample during one training iteration.

Algorithm 1 Zero-value masking (ZM)

Input: The hidden state of the target sample $x \in \mathbb{R}^{T \times F}$; The number of the consecutive time frames t; The starting time index t_0; The number of the consecutive frequency channels f; The starting frequency index f_0;
Output: The augmented hidden state of the target sample $x' \in \mathbb{R}^{T \times F}$;

$x' = x$;
for $i = t_0, \ldots, t_0 + t$ **do**
  for $j = 0, \ldots, F$ **do**
    $x'[i, j] = 0$;
  end for
end for
for $i = 0, \ldots, T$ **do**
  for $j = f_0, \ldots, f_0 + f$ **do**
    $x'[i, j] = 0$;
  end for
end for
return x';

Algorithm 2 Mini-batch based mixture masking (MM)

Input: The hidden state of the target sample $x \in \mathbb{R}^{T \times F}$; The hidden state of another sample within the same mini-batch $y \in \mathbb{R}^{T \times F}$; The number of the consecutive time frames t; The starting time index t_0; The number of the consecutive frequency channels f; The starting frequency index f_0;
Output: The augmented hidden state of the target sample $x' \in \mathbb{R}^{T \times F}$;

$x' = x$;
for $i = t_0, \ldots, t_0 + t$ **do**
  for $j = 0, \ldots, F$ **do**
    $x'[i, j] = \frac{1}{2}(x[i, j] + y[i, j])$;
  end for
end for
for $i = 0, \ldots, T$ **do**
  for $j = f_0, \ldots, f_0 + f$ **do**
    $x'[i, j] = \frac{1}{2}(x[i, j] + y[i, j])$;
  end for
end for
return x';

2.2. Masking Schemes

As shown in Figure 1, we present three masking schemes, including the zero-value masking (ZM), the mini-batch based mixture masking (MM) and the mini-batch based cutting masking (CM). ZM directly applies the zero value for the masking regions of the target sample. MM and CM utilize the time frames and frequency channels from another sample for masking. To explain, if the hidden state in the l-th layer of the target sample is to be augmented, we randomly select another sample within the same mini-batch as the target sample and use the hidden state in the l-th layer of the selected sample for masking. The difference is that MM mixes the masking regions of the hidden states of the two samples by the mean, while CM fills the masking regions of the target sample with the same regions of the selected sample. The details are summarized in Algorithm 1,

Algorithm 3 Mini-batch based cutting masking (CM)

Input: The hidden state of the target sample $x \in \mathbb{R}^{T \times F}$; The hidden state of another sample within the same mini-batch $y \in \mathbb{R}^{T \times F}$; The number of the consecutive time frames t; The starting time index t_0; The number of the consecutive frequency channels f; The starting frequency index f_0;
Output: The augmented hidden state of the target sample $x' \in \mathbb{R}^{T \times F}$;

$x' = x$;
for $i = t_0, \ldots, t_0 + t$ **do**
  for $j = 0, \ldots, F$ **do**
    $x'[i, j] = y[i, j]$;
  end for
end for
for $i = 0, \ldots, T$ **do**
  for $j = f_0, \ldots, f_0 + f$ **do**
    $x'[i, j] = y[i, j]$;
  end for
end for
return x';

Algorithm 2 and Algorithm 3, respectively. We further discuss their pros and cons in Section 4.3.

3. Models

We use CP-ResNet[1] [23] as the base model for the ASC task. CP-ResNet is a variant of ResNet [24] by adapting the audio tasks using receptive field (RF) regularization, which shows the best performance among single models for ASC [25]. The network architecture is detailed in Table 1. There are a series of residual convolutional blocks with a kernel size of 3×3 or 1×1, followed by a global average pooling function and a fully-connected layer. CP-ResNet has a RF of 87×87 and a total of $6,053,780$ trainable parameters.

Our proposed method is applied to the CP-ResNet between the model blocks. More specifically, the augmentation is applied to the input log mel spectrograms (denoted as Layer 0), the hidden states before residual block (RB) 1 (denoted as Layer 1), before RB 5 (denoted as Layer 2), before RB 9 (denoted as Layer 3) and after RB 12 (denoted as Layer 4), respectively. Hence, both the input space (Layer 0) and the hidden space (Layers 1-4) can be enhanced.

4. Experiments

4.1. Datasets

The experiments were conducted on the DCASE 2018 [21] and DCASE 2019 [22] ASC task 1a datasets, which are commonly used benchmark datasets for ASC. The DCASE 2018 task 1a dataset [21] consists of about 17 hours of audio for training (6122 10-second clips) and 7 hours for evaluation (2518 10-second clips). The DCASE 2019 task 1a dataset [22] contains a total of 40 hours of audio (9185 clips in the training set, 4185 clips in the test set). Each recording belongs to one out of 10 possible classes.

4.2. Setups

For all experiments, we follow the same settings as the state of the art [23, 26] to ensure a fair comparison. Specifically, the input is down-sampled to 22.05kHz and applied a Short Time Fourier Transform (STFT) with a window size of 2048 and 25% overlap, followed by a Mel-scaled filter bank on perceptually weighted spectrograms. This results in 256 Mel frequency bins and around 43 frames per second. The input frames are normalized to zero-mean and unit variance according to the training set. The Adam optimizer [27] is used for a total of 350 epochs, with an initial learning rate of 1×10^{-4}. The learning rate decays linearly from epoch 50 until 250 where it reaches 5×10^{-6}. Then we train for another 100 epochs with the minimum learning rate 5×10^{-6}. The models are evaluated on the test set after 350 epochs of training. Each experiment is repeated 3 times, and we report the mean and the standard deviation of these runs. When using SpecAugment++[2], we randomly select a layer to perform the augmentation from a set of layers (Layers 0-4). Unless otherwise stated, we set the hyperparameters t' as 43 and f' as 26 in our experiments. Thus, around 10% of the time frames and frequency channels are masked.

Table 1: *Network architecture of CP-ResNet*

RB Number	RB Config
	5×5, C=64, S=2
1	3×3, 1×1, C=128, S=1, P
2	3×3, 3×3, C=128, S=1, P
3	3×3, 3×3, C=128, S=1
4	3×3, 3×3, C=128, S=1, P
5	3×3, 1×1, C=256, S=1
6	1×1, 1×1, C=256, S=1
7	1×1, 1×1, C=256, S=1
8	1×1, 1×1, C=256, S=1
9	1×1, 1×1, C=512, S=1
10	1×1, 1×1, C=512, S=1
11	1×1, 1×1, C=512, S=1
12	1×1, 1×1, C=512, S=1

RB: Residual Block; S: stride; P: 2×2 max pooling after the block; C: Number of channels.

Table 2: *Comparisons of classification accuracy on different data dugmentation methods (%).*

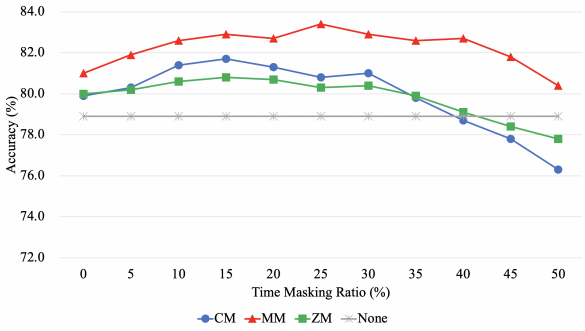
Augmentation Method	DCASE 18	DCASE 19
No augmentation [23]	74.3 ± 0.59	78.9 ± 0.80
mixup (2017) [15]	75.5 ± 0.62	79.3 ± 0.71
SpecAugment (2019) [14]	74.9 ± 0.81	79.1 ± 1.05
BC Learning (2017) [17]	75.8 ± 0.66	80.0 ± 0.76
SpeechMix (2020) [18]	75.8 ± 0.48	80.7 ± 0.69
SpecAugment++ (ZM)	76.2 ± 0.59	80.6 ± 0.82
SpecAugment++ (MM)	$\mathbf{77.0 \pm 0.52}$	$\mathbf{82.6 \pm 0.66}$
SpecAugment++ (CM)	76.9 ± 0.73	81.4 ± 0.94

4.3. Results

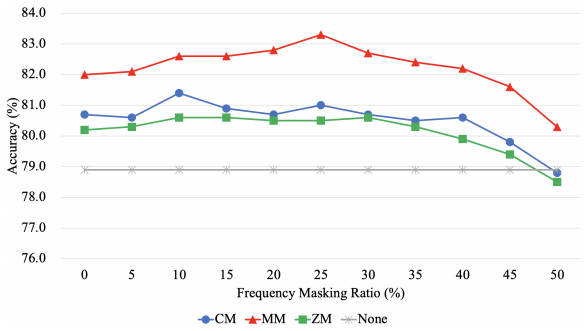
We compare the performance of our proposed method with the state of the art [14, 15, 17, 18], and the results are summarized in Table 3. Under a strong baseline model [23], the proposed SpecAugment++ significantly improves the performance on both datasets, and outperforms the other previous data augmentation methods (*i.e.* mixup [15], SpecAugment [14], BC

[1] https://github.com/kkoutini/cpjku_dcase19/

[2] https://github.com/WangHelin1997/SpecAugment-plus



(a) Comparison of different time masking ratios.



(b) Comparison of difference frequency masking ratios.

Figure 2: *Accuracy comparison of difference time masking ratios and frequency masking ratios on the DCASE 19. Here, we report the mean of three experiments. To evaluate the influence of time masking, we keep a frequency masking ratio of* 10%. *Similarly, a time masking ratio of* 10% *is kept when comparing different frequency masking ratios.*

Learning [17] and SpeechMix [18]). Among them, SpecAugment++ with MM achieves the highest accuracy gain (3.6% on DCASE 18 and 4.7% on DCASE 19), followed by SpecAugment++ with CM (3.5% on DCASE 18 and 3.2% on DCASE 19) and SpecAugment++ with ZM (2.6% on DCASE 18 and 2.2% on DCASE 19). These feature masking schemes improve generalization and robustness by enabling a model to attend not only to the most discriminative time frames and frequency channels, but also to the entire temporal and frequency parts.

Comparison of ZM, MM and CM. SpecAugment++ with ZM can be viewed as a generalization of SpecAugment [14], where the zero-value masking is applied to both the input log mel spectrograms and the hidden states of the network. ZM can let a model focus on less discriminative time frames and frequency channels, while not being efficient enough due to unused parts of the audio. MM and CM performs better than ZM because the time frames and frequency channels are masked by the corresponding parts of other audio, which introduce the interference and force the network to be more discriminative. However, CM may cause unnatural artifacts due to the abruptness of the masking parts and the original hidden states. In comparison, MM is a more natural choice, as the information of the whole target audio sample is retained and the interference of time frames and frequency channels from other audio samples can enhance the robustness. In addition, compared with mixup [15], BC Learning [17] and SpeechMix [18] which also mix the audio samples, ZM and CM do not change the label of the target audio sample so that the training is more stable.

Table 3: *Ablation analysis of the layers set used for the augmentation. While applying the SpecAugment++, we randomly select a layer to perform the time masking and frequency masking from the set of layers. Layer* 0 *denotes the input log mel spectrogram and Layers* 1-4 *denote the hidden states, which have been described in Section 3. Here, we report the mean of the three experiments on the DCASE 19 (%).*

		CM	MM	ZM
Layers Set	-	78.9	78.9	78.9
	$\{0\}$	80.5	81.8	79.8
	$\{1\}$	80.1	81.2	79.6
	$\{2\}$	80.2	81.0	79.4
	$\{3\}$	79.7	80.6	79.5
	$\{4\}$	79.9	80.7	79.3
	$\{0, 1\}$	80.9	82.1	80.0
	$\{0, 1, 2\}$	81.1	82.4	80.5
	$\{0, 1, 2, 3\}$	81.2	82.3	80.6
	$\{0, 1, 2, 3, 4\}$	81.4	82.6	80.6

4.4. Ablation Study

In order to explore the effectiveness of the time masking and frequency masking, we compare different time masking ratios and frequency masking ratios and show the results in Figure 2. It can be seen that both time masking and frequency masking improve the performance. As the ratios of time masking and frequency masking increase, the accuracy of all the three types of masking (*i.e.* ZM, MM and CM) is boosted and then drops, which shows the optimal performance when the ratio is $[10\%, 25\%]$. We argue that a masking ratio of around 10% is enough to guide a model focus on less discriminative time frames and frequency channels and increase the robustness. However, if the masking ratio is too large (*e.g.* over 40%), the information of the original audio may be lost significantly, which confuses the model severely and makes the model hard to be trained.

In addition, we investigate different sets of layers for SpecAugment++ on the DCASE 19 dataset, and results are shown in Table 3. When no augmentation is performed, the model accuracy is 78.9%. The performance tends to improve with the augmentation both on the input spectrogram (Layer 0) and on the hidden states (Layers 1-4), and the improvement is more significant when we apply the SpecAugment++ to the layers near the input. These layers learn discriminative features such as frequency response which is quite similar to the human perception [17]. Our proposed method achieves the best performance when used to all the layers.

5. Conclusions

We have presented a hidden space data augmentation method for acoustic scene classification, which can broaden the intermediate feature representations for deep neural networks. Our proposed method is simple and computationally cheap to apply, which has shown better performance than the state-of-the-art methods.

6. Acknowledgements

This paper was partially supported by the Shenzhen Science & Technology Fundamental Research Programs (No: JCYJ20180507182908274 & JSGG20191129105421211) and GXWD20201231165807007-20200814115301001.

7. References

[1] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for DCASE challenge 2016," *Proceedings of Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[2] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016, pp. 95–99.

[3] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the DCASE 2017 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, 2019.

[4] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 559–563.

[5] H. Wang, Y. Zou, and D. Chong, "Acoustic scene classification with spectrogram processing strategies," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 210–214.

[6] H. Wang, Y. Zou, D. Chong, and W. Wang, "Modeling label dependencies for audio tagging with graph convolutional network," *IEEE Signal Processing Letters*, vol. 27, pp. 1560–1564, 2020.

[7] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6440–6444.

[8] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[9] D. Zou and Q. Gu, "An improved analysis of training overparameterized deep neural networks," *Advances in Neural Information Processing Systems*, 2019.

[10] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: learning sound representations from unlabeled video," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 892–900.

[11] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2721–2725.

[12] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[13] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[14] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech*, pp. 2613–2617, 2019.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[16] H. Wang, Y. Zou, and W. Wang, "A global-local attention framework for weakly labelled audio tagging," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 351–355.

[17] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *International Conference on Learning Representations*, 2018.

[18] A. Jindal, N. E. Ranganatha, A. Didolkar, A. G. Chowdhury, D. Jin, R. Sawhney, and R. R. Shah, "SpeechMix—augmenting deep sound recognition using hidden space interpolations," *Proc. Interspeech*, pp. 861–865, 2020.

[19] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," *arXiv preprint arXiv:1907.06639*, 2019.

[20] H. Wang, Y. Zou, D. Chong, and W. Wang, "Environmental sound classification with parallel temporal-spectral attention," in *Proc. Interspeech*, 2020, pp. 821–825.

[21] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 9–13.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019, pp. 164–168.

[23] 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 *27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] K. Koutini, F. Henkel, H. Eghbal-Zadeh, and G. Widmer, "Low-complexity models for acoustic scene classification based on receptive field regularization and frequency damping," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 86–90.

[26] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," in *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2018.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.