



# Convolutional Recurrent Neural Network with Auxiliary Stream for Robust Variable-Length Acoustic Scene Classification

Won-Gook Choi<sup>1</sup>, Joon-Hyuk Chang<sup>1</sup>

<sup>1</sup>Department of Electronic Engineering  
Hanyang University, Seoul, Republic of Korea

onlyworld94@naver.com, jchang@hanyang.ac.kr

## Abstract

Deep learning has proven to be suitable for acoustic scene classification (ASC). Therefore, it exhibits significant improvement in performance while using neural networks. However, several studies have been performed using convolutional neural network (CNN) rather than recurrent neural network (RNN) or convolutional recurrent neural network (CRNN), even though acoustic scene data is treated as a temporal signal. In practice, CRNNs are rarely adopted and are ranked lower in recent detection and classification of acoustic scenes and events (DCASE) challenges for fixed-length (i.e., 10 s) ASC. In this paper, an auxiliary stream technique is proposed that can improve the performance of CRNNs compared with that of CNNs by controlling the inductive bias of RNN. The auxiliary stream trains CNN by effectively extracting embeddings and is only connected on training steps. Therefore, it does not affect the model complexity on the inference steps. The experimental results demonstrate the superiority of the proposed method, regardless of the CNN model used for CRNN. Additionally, the proposed method yields robustness on variable-length ASC by performing streaming inferences and demonstrates the importance of CRNN.

**Index Terms:** acoustic scene classification, convolutional recurrent neural network, variable-length, streaming

## 1. Introduction

Acoustic scene classification (ASC) refers to the classification of an audio recording to detect the environment in which it was recorded by hearing the local soundscapes. An acoustic scene is determined using various factors such as sound events, reverberation, and a few acoustic noises [1, 2]. A dominant sound event or acoustic context of various considerations can determine the acoustic scene. ASC plays a key role in machine hearing and has been adopted as the main challenge for several years in the detection and classification of acoustic scenes and events (DCASE) challenges [3–5]. Recent studies have focused on constructing the ASC models using neural network-based systems such as convolutional neural network (CNN) [6], convolutional recurrent neural network (CRNN) [7], and transformer-augmented networks [8]. The conventional strategy to learn the patterns of acoustic scene data explores a time-frequency feature using a log-mel spectrogram. In particular, CNN-based architectures are being extensively investigated for ASC because they can extract high-level feature maps. Indeed, in the DCASE challenges over the recent years, CNN-based systems highly ranked in every ASC task, classifying ten acoustic scenes with 10 s length audio clips [9–13].

Although an acoustic scene is a time-series data, recent studies have used CNN-based architectures instead of CRNN-based architectures that can extract high-level time series em-

beddings and determine the context. Three teams [7, 14, 15] adopted the CRNN architecture for ASC tasks (among the 130 DCASE challengers for three years), but failed to record a higher rank. CRNN architectures have been effectively used for neural network-based acoustic tasks such as sound event detection (SED) [16]. However, they are rarely used for ASC due to their low performance. CNN can effectively extract information-intensive feature maps from a time-frequency feature. Still, there exists a restriction on the variable-length scene awareness considering a limitation of the receptive fields (RF) of CNN [17]. In this regard, the hidden states of RNN can compensate for the limitation of the RF. However, the inductive bias of RNN might interfere with extracting embeddings through CNN and leads to low performance.

This study purposes to improve the performance of CRNN than that of CNN regardless of the length of the input sequence by adopting the auxiliary stream to sequential embeddings. The embeddings extracted by CNN are trained parallel on both the main stream (i.e., RNN) and auxiliary stream. The proposed method for ASC does not require additional labeling such as onset and offset labels of sound events to build an auxiliary stream. In other words, only the loss for training the embeddings is calculated on the stream using frame-wise scene labels. Since the outputs of the stream do not affect the final inference (i.e., the outputs of RNN), the stream is only activated during the training steps, and the number of parameters is sustained on inference steps as that of vanilla CRNN.

Experiments are conducted with various CNN structures on the fixed-length (i.e., 10 s) audio clips to prove the effectiveness of the proposed method. Subsequently, streaming inferences are performed to verify the performance of the variable-length input sequence. Considering the capacity limitations of the mobile devices for the ASC service, experiments are conducted while maintaining the number of model parameters low enough. Through experiments, we show that the auxiliary loss reduces the effect of RNN on CNN while training the CRNN model, and the auxiliary stream-augmented CRNN outperforms the CNN-based model while maintaining the same model complexity as that of the vanilla CRNN that yields a poor performance than that of CNN.

## 2. Backgrounds

### 2.1. Backpropagation Through Time

Considering the backpropagation through time (BPTT) [18] of RNN, the gradient of loss  $E$  for the  $i$ -th  $d$ -dimensional embedding of the  $n$ -sequence  $\{e_i^d\}_{i=1}^n$  is expressed as

$$\frac{\partial E}{\partial e_i^d} = \frac{\partial E}{\partial \mathbf{y}_n} \frac{\partial \mathbf{y}_n}{\partial \mathbf{h}_n} \frac{\partial \mathbf{h}_n}{\partial \mathbf{h}_i} \frac{\partial \mathbf{h}_i}{\partial e_i^d}, \quad (1)$$

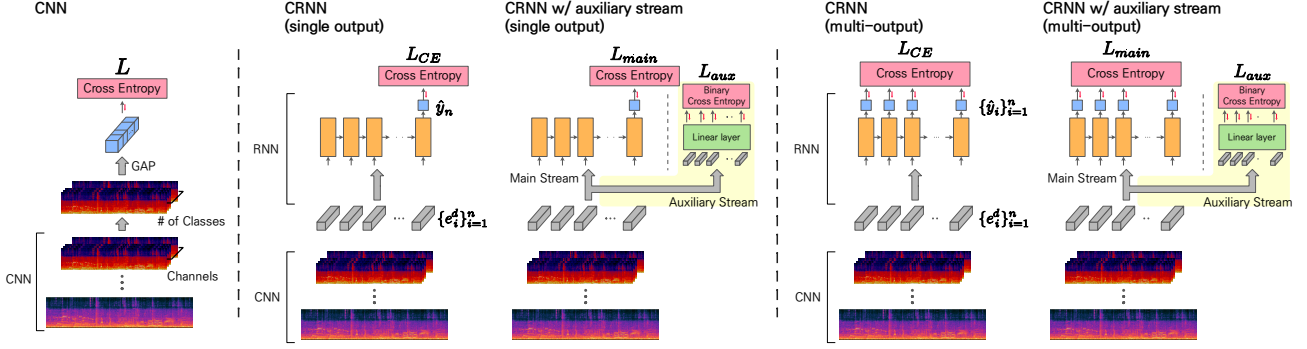


Figure 1: Overall architectures of CNN, CRNN, and CRNN with auxiliary stream.

where  $\hat{y}_n$  and  $h_i$  denote the probability of the last output of the RNN and the  $i$ -th hidden state of the recurrent layer, respectively ( $\frac{\partial \hat{y}_n}{\partial h_n} = W_{hy}^T$ ,  $\frac{\partial h_i}{\partial e_i^d} = W_{hx}^T$ ,  $\frac{\partial h_n}{\partial h_{n-1}} = W_{hh}^T$ ) if activation functions and biases are neglected. When optimizing the parameters of CNN, they depend upon the geometric series of  $W_{hh}$  (i.e., the term  $\sum_{i=1}^n \frac{\partial h_n}{\partial h_i}$ ), which is a hidden layer for satisfying the following hypothesis with the  $i$ -th target  $y_i$ :

$$P(y_i | e_1, \dots, e_{i-1}, e_i) = P(y_i | h_{i-1}, e_i). \quad (2)$$

It might make CNNs less trained about the classification information because of the training under the inductive-bias: sequential-attributes. This study introduces the strategy that extracts embeddings important for scene classification by adding a parallel stream that does not pass through RNN.

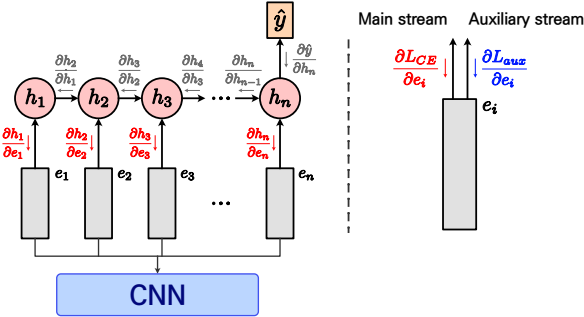


Figure 2: Illustration of backpropagation through time.

## 2.2. Variable-Length ASC

Seo *et al.* [17] proposed the shallow Conformer for robust variable-length ASC and it outperformed CNN while training using a short-length audio clip (i.e., 1–3 s) and evaluating for 1–10 s. They conducted training and testing at varying lengths of audio signals to evaluate robust variable-length ASC. In this study, it is considered that variable-length ASC is required for streaming inference. Therefore, the robustness of the variable-length ASC are evaluated slightly differently. The model is being training at 10 s, whereas evaluating with streaming inference is performing in the range of 1–10 s. The performance enhances as the input length grows, but the computational cost and memory demand also increase. Hence, it is considerably advantageous if the system has robustness on short sequences, even though it is trained using long sequences. In this regard, the proposed method for CRNN has significant benefits.

## 3. Proposed Method

In this section, the CRNN architecture widely used in acoustic-based deep learning research is firstly reviewed. Additionally, the steps performed to design the auxiliary stream with the auxiliary loss are introduced. Vanilla CRNN [7] without special techniques is adopted to prove the superiority of the auxiliary stream.

### 3.1. CRNN Architecture

Letting  $X$  be the time-frequency features extracted from an audio signal,  $f_d(X)$  is the high-level feature maps that CNN detects from the  $X$ , where  $f_d(\cdot)$  denotes the function of  $d$ -channel CNN. Subsequently, sequential embeddings are extracted from the feature maps to obtain sequential features and information using RNN. Conventionally, the feature maps are connected to a fully-connected layer on the frequency axis frame-by-frame to ensure that  $\{e_i^d\}_{i=1}^n$  are generated. Since RNN has internal memories called hidden states, the neural network can observe the past information,  $h_{i-1}$ , which satisfies Eq. (2) under the inductive bias of RNN, when  $e_i^d$  is fed to RNN. RNN brings out the inferences  $\{\hat{y}_i\}_{i=1}^n$  and the final output  $\hat{y}_n$  is obtained while calculating the loss with respect to the ground truth  $y$ , if a single-output RNN is assumed (Fig. 1). In this study, single-output and multi-output CRNNs are adopted. In the case of multi-output CRNNs, the loss  $L_{CE}$  is calculated as the mean of each cross-entropy for each inference.

### 3.2. Auxiliary Stream

CNN extracts the embedding sequences from the log-mel spectrogram in the CRNN architecture. A stream is constructed that is directly headed to ground truth (not through the RNN) named auxiliary stream and the final inference is only decided on the main stream separating from the auxiliary stream. Therefore, the auxiliary stream is not used on inference steps as displayed in Fig. 1. The embedding vectors  $\{e_i^d\}_{i=1}^n$  pass through the linear layer  $W_{aux} \in \mathbb{R}^{d \times c}$  on the auxiliary stream that transforms the  $d$ -dimension vector to the  $c$ -dimension classes. Assuming a 10 s fixed-length acoustic scene audio without scene-transition, each embedding is extracted from each bundle of frames in the log-mel spectrogram of the same scene. Hence, the embeddings from an audio segment have the same label as that of the ground truth. This property differentiates ASC from tasks such as SED that has different labels on each frame. Therefore, the model can efficiently extract the feature maps through a comparison between each embedding and the ground truth as shown

in Fig. 1. Each vector is then scored by the sigmoid as follows:

$$\tilde{y}_i = \sigma(W_{aux}^T e_i^d + b_{aux}) \quad (3)$$

where  $\sigma(\cdot)$  and  $b_{aux}$  denotes the sigmoid function and the bias of linear layer, respectively. Then, the auxiliary loss  $L_{aux}$  is calculated by the binary cross-entropy (BCE) loss function:

$$L_{aux} = -\frac{1}{nc} \sum_{i=1}^n \sum_{l=1}^c (y_l \ln \tilde{y}_{il} + (1 - y_l) \ln(1 - \tilde{y}_{il})). \quad (4)$$

Empirically, it is observed that the BCE is relatively suitable for extracting embedding vectors rather than cross-entropy (CE) for the auxiliary stream. The total loss  $L_{total}$  for training the network is calculated as the sum of  $L_{CE}$  and  $L_{aux}$ .

$$L_{total} = L_{CE} + L_{aux}. \quad (5)$$

## 4. Experiments and Results

### 4.1. Dataset

There are different versions of the TAU urban acoustic scenes dataset (TAU) depending on the purposes [19–23]. Unlike the TAU 2020 mobile dataset [20], which was used for the DCASE 2021 challenge aiming at the classification of scenes considering generalization across several different devices, we employed the TAU 2019 dataset [19] to concentrate on the variable-length ASC without considering domain adaptation problems about mismatched or unseen devices. Each audio file consisted of real recordings of 10 s collected on ten acoustic scenes in twelve cities with 48 kHz, two-channel. Audio samples were down-mixed to 16 kHz and processed to generate log-mel spectrograms with window size, shift size, and mel bins of 128 ms, 32 ms, and 256 bins, respectively. The total number of segments was 14,400 and the training and validation segments were divided into 9,185 and 4,185 segments, respectively.

### 4.2. Model and Training

Three CNN models were adopted for building CRNN, which exhibits good performance in the ASC system: ResNet [24], BCResNet [25], and FUSE [26]. The number of parameters in the CNNs was adjusted to be similar (about 100,000–150,000) and the same RNN architecture was used with one-sided gated recurrent units (GRU), two layers and ten hidden sizes. The classifiers of CNNs (i.e., the conventional CNNs for comparison, not those for the auxiliary stream, Fig. 1) has the same

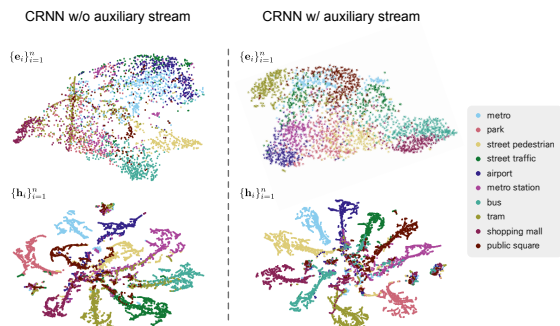


Figure 3: *tSNE distributions of embedding sequence  $\{e_i\}_{i=1}^n$  and hidden states  $\{h_i\}_{i=1}^n$ . The distributions of hidden states looked diverged from the center with the progression of steps.*

algorithm: adjusted the number of the last feature maps to the number of classes and passed them to the global average pooling layer that is widely adopted in the ASC task including the state-of-the-art models on the DCASE 2020 [10] and 2021 [9] challenge. This ensured that the CNN model could receive variable-length audio as the input while retaining the same architecture [17]. The training epochs, learning rate, and weight decay used in the models were 50, 0.05, and 0.001, respectively. The cross entropy loss at the outputs of RNN and cosine annealing with scheduler with stochastic gradient descent optimizer was used to optimize the models, and the initial 5 epochs were warmed up.

Table 1: *Accuracy for various CNN and CRNN architectures. Every performance was averaged on five random seeds. For validation on 1 s performances, we randomly sampled from each audio file.*

CNN Architecture	RNN Architecture	Accuracy		
		10 s	1 s	
<b>ResNet</b>	CNN only Global Average Pooling	70.56%	52.28%	
	# num. of blocks = {3, 3, 3, 1}, stacks = {2, 2, 2, 2}, filters = {8, 16, 32, 64}	GRU (single output) w/o auxiliary stream	66.49%	44.22%
		GRU (single output) <b>w/ auxiliary stream</b>	<u>74.75%</u>	<u>60.50%</u>
	# num. of params. CNN = 127.5K CRNN = 136.8K	GRU (multi-output) w/o auxiliary stream	67.51%	59.68%
	GRU (multi-output) <b>w/ auxiliary stream</b>	<b>74.82%</b>	<b>62.33%</b>	
<b>BCResNet</b>	CNN only Global Average Pooling	75.34%	53.79%	
	# num. of channels = 50	GRU (single output) w/o auxiliary stream	69.10%	52.24%
		GRU (single output) <b>w/ auxiliary stream</b>	<b>76.91%</b>	<b>64.18%</b>
	# num. of params. CNN = 130.0K CRNN = 154.9K	GRU (multi-output) w/o auxiliary stream	72.68%	61.22%
	GRU (multi-output) <b>w/ auxiliary stream</b>	<u>76.80%</u>	<b>66.37%</b>	
<b>FUSE</b>	CNN only Global Average Pooling	74.38%	49.35%	
	# num. of channels = 32, blocks = 3	GRU (single output) w/o auxiliary stream	72.74%	52.76%
		GRU (single output) <b>w/ auxiliary stream</b>	<u>76.10%</u>	<u>64.58%</u>
	# num. of params. CNN = 130.6K CRNN = 147.8K	GRU (multi-output) w/o auxiliary stream	74.82%	63.09%
	GRU (multi-output) <b>w/ auxiliary stream</b>	<b>77.76%</b>	<b>66.88%</b>	

### 4.3. Results

Experiments with various CNN models were performed to verify whether RNN deteriorated the classification performance of CNN and the auxiliary stream-augmented CRNN outperformed CNN. Table 1 demonstrates the results of the experiments for long (10 s) and short (1 s) audio signals. It demonstrated that CNN showed a better performance than that of CRNN in each architecture (accuracies of 10 s). Conversely, in the case of adding the auxiliary stream on the training steps, CRNN outperformed CNN for single-output and multi-output, while maintaining the model complexity same to that of the vanilla

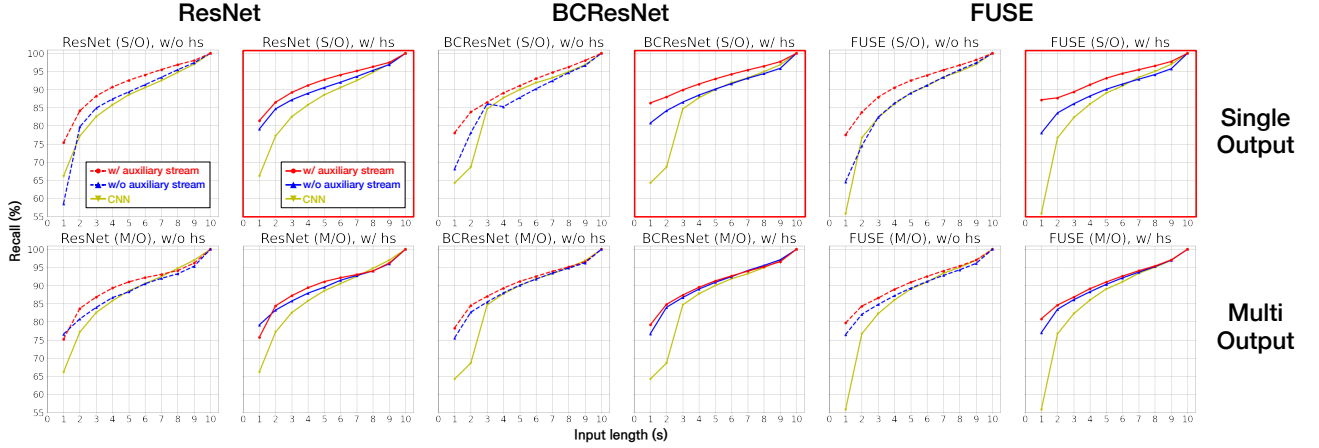
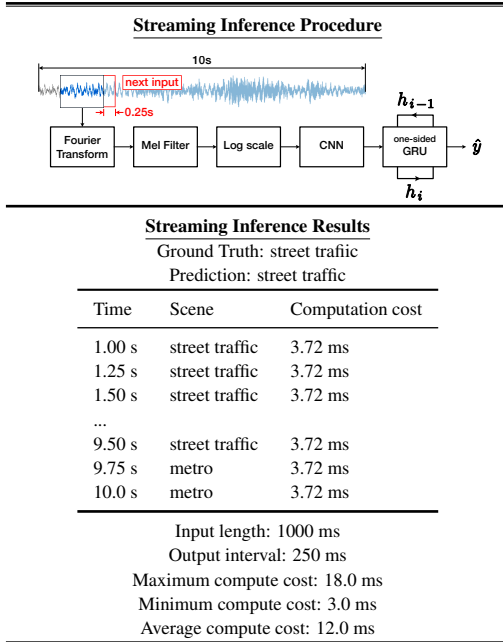


Figure 4: Recalls for the streaming inferences on one to ten secs of input audio length. (w/o hs: without hidden states, dashed-lines)

Table 2: Streaming inference procedure. If hidden states were not used (dashed line in Fig. 4), then  $h_i$  was zero for every  $i$ .



CRNNs.

The  $t$ -distributed stochastic neighbor embedding ( $t$ SNE) distributions of embedding sequences and hidden states were visualized in Fig. 3 to demonstrate the effect of adding the auxiliary stream. In Section 2.1, it was assumed that the auxiliary loss increases the classification accuracy because the embedding sequence directly trains the classification information while controlling the inductive bias of RNN. As shown in the figure, the embeddings and hidden states of the proposed method were well spatially and linearly clustered, respectively. It demonstrated that the auxiliary stream reduced the inductive bias of RNN at the embedding level and reinforced it at the hidden state level. As a result, we deduce that the auxiliary loss enhanced the operation of CNN and RNN.

Additionally, the proposed method was robust for 1 s audio signals, which are the extreme condition for variable-length

inference as summarized in Table 1. A detailed validation of the variable-length system was obtained by performing streaming inferences on the validation segments by controlling the input length in the range of 1–10 s, with an inference interval of 0.25 s as shown in Table 2. Subsequently, each variable-length classification performance was recorded on the plot. Fig. 4 reports the recall for the result of each inference step when true positive audio segments were provided as the input data for the 10 s fixed-length inference (i.e., it was ensured that the systems worked perfectly on 10 s ASC). It can be observed that the auxiliary stream-added CRNN was relatively robust on the variable-length ASC. The improved CRNN was robust on variable-length when they did not use the hidden states. However, as described in Section 1 and 2.2, CRNN had significant advantages on variable-length (especially for shorter input sequences) and streaming inference compared with that of CNNs while using the hidden states.

## 5. Conclusions

In this paper, an auxiliary stream-augmented CRNN architecture for ASC was proposed that exhibited robustness on the variable-length audio. The auxiliary loss makes the embedding sequences contain more information for classification by getting lower inductive bias of RNN. The experimental results demonstrated that the proposed method outperformed CNN and the vanilla CRNN. In addition, the results of streaming inferences were used to prove the importance of CRNN in variable-length ASC, and the system exhibited robustness with a decrease in the input sequence. For the future works, we will aim to design auxiliary streams for the variable-length ASC with mismatched and unseen device problems, which the streams will be designed to extract the embeddings with generalizing the device properties.

## 6. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

## 7. References

- [1] A. Hüwel, K. Adiloğlu, and J. Bach, “Hearing aid research data set for acoustic environment recognition,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 706–710.
- [2] M. Kośmider, “Spectrum correction: Acoustic scene classification with mismatched recording devices,” in *Proc. Interspeech*, 2020, pp. 4641–4645.
- [3] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, November, 2018, pp. 9–13.
- [4] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase challenge: generalization across devices and low complexity solutions,” in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 56–60.
- [5] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, “Low-complexity acoustic scene classification for multi-device audio: analysis of dcase challenge systems,” *arXiv:2105.13734*, 2021.
- [6] 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 845–849.
- [7] E. Fanioudakis and A. Vafeiadis, “Investigating temporal and spectral sequences combining GRU-RNNs for acoustic scene classification,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2020.
- [8] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio spectrogram transformer,” in *Proc. Interspeech*, 2021, pp. 571–575.
- [9] B. Kim, S. Yang, J. Kim, and S. Chang, “QTI submission to DCASE 2021: residual normalization for device-imbalanced acoustic scene classification with efficient design,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2021.
- [10] S. Suh, S. Park, Y. Jeong, and T. Lee, “Designing acoustic scene classification models with cnn variants,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2020.
- [11] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, “CP-JKU submissions to DCASE’20: Low-complexity cross-device acoustic scene classification with RF-regularized CNNs,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2020.
- [12] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, “Integrating the data augmentation scheme with various classifiers for acoustic scene modeling,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2019.
- [13] M. Kośmider, “Calibrating neural networks for secondary recording devices,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2019.
- [14] L. Pham, T. Doan, D. T. Ngo, H. Nguyen, and H. H. Kha, “CDNN-CRNN joined model for acoustic scene classification,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2019.
- [15] H. Zhu, C. Ren, J. Wang, S. Li, L. Wang, and L. Yang, “Dcase 2019 challenge task1 technical report,” *the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, Tech. Rep., 2019.
- [16] 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.
- [17] S. Seo, D. Lee, and J.-H. Kim, “Shallow convolution-augmented transformer with differentiable neural computer for low-complexity classification of variable-length acoustic scene,” in *Proc. Interspeech*, 2021, pp. 576–580.
- [18] T. P. Lillicrap and A. Santoro, “Backpropagation through time and the brain,” *Current Opinion in Neurobiology*, vol. 55, pp. 82–89, 2019.
- [19] T. Heittola, A. Mesaros, and T. Virtanen, “TAU urban acoustic scenes 2019, development dataset.” [Online]. Available: <https://doi.org/10.5281/zenodo.2589280>
- [20] —, “TAU urban acoustic scenes 2020 mobile, development dataset.” [Online]. Available: <https://doi.org/10.5281/zenodo.3819968>
- [21] —, “TAU urban acoustic scenes 2020 3class, development dataset.” [Online]. Available: <https://doi.org/10.5281/zenodo.3670185>
- [22] —, “TAU urban acoustic scenes 2019 mobile, development dataset.” [Online]. Available: <https://doi.org/10.5281/zenodo.2589332>
- [23] —, “TAU urban acoustic scenes 2019 openset, development dataset.” [Online]. Available: <https://doi.org/10.5281/zenodo.2591503>
- [24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [25] B. Kim, S. Yang, J. Kim, and S. Chang, “Domain generalization on efficient acoustic scene classification using residual normalization,” in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, 2021, pp. 21–25.
- [26] W.-G. Choi, J.-H. Chang, J.-M. Yang, and H.-G. Moon, “Instance-level loss based multiple-instance learning for acoustic scene classification,” *arXiv:2203.08439*, 2022.