Acoustic Scene Analysis with Multi-head Attention Networks

Weimin Wang* *Weiran Wang*† *Ming Sun** *Chao Wang**

*Amazon Alexa †Salesforce Research

{wanweimi,mingsun,wngcha}@amazon.com weiran.wang@salesforce.com

Abstract

Acoustic Scene Classification (ASC) is a challenging task, as a single scene may involve multiple events that contain complex sound patterns. For example, a cooking scene may contain several sound sources including silverware clinking, chopping, frying, etc. What complicates ASC more is that classes of different activities could have overlapping sounds patterns (e.g. both cooking and dishwashing could have silverware clinking sound). In this paper, we propose a multi-head attention network to model the complex temporal input structures for ASC. The proposed network takes the audio's time-frequency representation as input, and it leverages standard VGG plus LSTM layers to extract high-level feature representation. Further more, it applies multiple attention heads to summarize various patterns of sound events into fixed dimensional representation, for the purpose of final scene classification. The whole network is trained in an end-to-end fashion with backpropagation. Experimental results confirm that our model discovers meaningful sound patterns through the attention mechanism, without using explicit supervision in the alignment. We evaluated our proposed model using DCASE 2018 Task 5 dataset, and achieved competitive performance on par with previous winner's results.

Keywords: acoustic scene analysis, unsupervised alignment learning, multi-head attention

1. Introduction

High level semantic understanding of an audio stream is a fundamental problem in machine intelligence. Being able to infer from sound patterns what events are occuring and what is the surrounding environment has potential applications in a wide range of fields such as public safety [1], ecological study [2], and assisted living [3]. Recently, the tasks of audio event detection (AED) and acoustic scene analysis (ASC) have gained increasing popularity, due to the availability of large scale datasets [4] and commonly used benchmarks [5, 6]. We observe that, deep learning architectures such as convolutional neural networks [7, 8] and long short-term memory networks [9] and their variants have contributed significantly to the success of many approaches to the tasks.

Although the common deep learning architectures can work well for fully supervised tasks, challenges arise when the task is only weakly labeled, which is often the case in practice. As an example, a *cooking* scene may contain several sound events including silverware clanking, chopping, frying, and perhaps other human activities (such as walking and talking); knowing the existence of the smaller events (with short temporal duration) clearly helps inferring the abstract scene class. To train a ASC system, it is challeging to collect datasets with fine-grained event labels: annotators may quickly categorize the whole scene correctly, but it would be laborious (and also expensive) for them to exhaustively identify the smaller sound events and to pinpoint their onset/offset times. A more practical approach is to collect sufficient amount of recordings with only scene class, and develop models which exploit the structure that a scene typically consists of multiple smaller events, and perform recording-level inference based on the aggregation of evidence at the event level, with supervision only at the scene level. Due to both the challenges and opportunities, weakly supervised learning has been a continual scheme in this area.

In this paper, we propose a multi-head attention network for ASC, which implements the abovementioned intuitions of hierarchical representation and compositional inference. Our model applies multiple attention heads to frame level representations of the input recording, where each head has a hidden event in mind, and attends to relevent frames to extract a recording-level features; the features from all attention heads are then pooled together, as the final representation of the entire recording for scene classification. Although we only receive supervision at the recording level, experimental results show that our model discovers meaningful sound patterns through the attention mechanism, and the soft alignment provided by the attention heads encode high-quality time information. In the rest of this paper, we give detailed formulation of our method in Sec. 2, discuss related work in Sec. 3, present experimental results in Sec. 4 and qualitative analysis of the attentions in Sec. 5, and conclude in Sec. 6.

2. Multi-head attention for ASC

Our task is to associate each audio clip with a scene class. The scene can be a high level, abstract concept which consist of various smaller events. For example, in a typical "cooking" scene, we expect to hear events like cook-ware, cutting, dishwashing, and human activity sounds like walking and talking. In another scene "working", we could hear events such as keyboard typing and mouse clicks, as well as paper scratching.

In order to categorize the overall scene, it can be helpful to detect the existence of such smaller events and to analyze their co-occurrence. However, it is very costly to pre-define the set of smaller events and have human annotate their occurrence in audio clips. In this section, we propose a method for automatically detecting the existence of meaningful events and locating their appearances in time (alignment) for scene classification.

Let an input utterance be $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{T'}]$ where $\mathbf{x}_t \in \mathbb{R}^d$ contains the features for the tth audio frame, and T' is the total number of frames. We apply deep convolutional networks followed by bi-directional LSTMs to extract high-level features that contain rich context information from the input (see Section 4.2 for details). Let the output of this feature extraction networks (denoted by f) be $f(\mathbf{X}) = [\mathbf{h}_1, \dots, \mathbf{h}_T]$ where $\mathbf{h}_t \in \mathbb{R}^p$ and $T \leq T'$ due to subsampling in the time axis.

For each input sequence, we consider a set of M smaller events, where M is a hyper-parameter to be tuned by cross-validation. Let the vectorial representation of the ith event be

Work done while Weiran Wang was at Amazon Alexa.

$\mathbf{v}^i \in \mathbb{R}^p$, and write representations of all events collectively as $\mathbf{V} = \{\mathbf{v}^i\}_{i=1,\dots,M}$. We compute the similarity between the $\mathbf{h}_t$ sequence and $\mathbf{v}^i$, followed by exponentiation and normalization, to obtain the "attention scores" for event i:

$$a_t^i = \exp(\mathbf{h}_t^\top \mathbf{v}^i/\sigma) / \sum_{t'=1}^{T} \exp(\mathbf{h}_{t'}^\top \mathbf{v}^i/\sigma), \quad \text{for } t = 1, \dots, T,$$

where $\sigma > 0$ is a hyper-parameter that controls the sharpness of the soft alignment (the smaller the σ is, the more peaked the attention scores are). The nonnegative attention scores $\{a_t^i\}_{t=1,\dots,T}$ satisfy $\sum_{t=1}^T a_t^i = 1$, and highlight the most relevant frames (for event i) from $f(X)$, while pushing the affinity of others frames close to zero. We then summarize the feature sequence into a fixed dimensional vector

$$\mathbf{s}^i = \sum_{t=1}^{T} a_t^i \mathbf{h}_t \in \mathbb{R}^p, \quad i = 1, \dots, M$$

for each event. Finally, we concatenate all M events' representations to obtain

$$\mathbf{s} = [\mathbf{s}^1; \dots; \mathbf{s}^M] \in \mathbb{R}^{Mp},$$

and use it as the final feature for the entire utterance.

For N-class scene classification, we apply a feed-forward network g with a final softmax layer to the utterance representation $\mathbf{s}$. with weights $\mathbf{W} \in \mathbb{R}^{N \times Mp}$ at the end, to get predictions

$$[P(y = 1|\mathbf{X}), \dots, P(y = N|\mathbf{X})] = \text{softmax}\,(g(\mathbf{s}))\,.$$

Given a training set of $(\mathbf{X}, y)$ pairs, we jointly learn parameters in feature extraction network f, event representations $\mathbf{V}$, and classification network g using the cross-entropy loss.

The attention mechanism is widely used in speech recognition [10, 11] and natural language processing [12], and multi-head self-attention has been proposed in [13]. We have borrowed the same intuition of learn-able, soft alignment from these prior work for detecting events in an unsupervised fashion, and our use of multiple attention heads is motivated by the complex nature of scenes—each scene may contain several distinctive events. In related settings, a few recent work [14, 15, 16] formulated the weakly supervised event detection problem (given only the utterance label, train a system to infer both utterance label and time alignment) as a multiple instance learning problem, and proposed different pooling strategies to aggregate the per-instance (or per-segment) hypothesis to form an utterance level prediction, on top of which supervision is imposed. This aggregation process resembles attention, albeit at the prediction score level for *specific* target event, rather than at the representation level. A similar attention mechanism to ours was used for rare event detection in [17], where the attention scores receives supervision from the onset/offset time provided by their task (Challenge 2 of DCASE 2017 [18]). In contrast, we do not have any supervision in this work for the frame-wise alignment. Instead, we rely on the model's structural constraint—the multi-head attention mechanism—to attend to multiple relevant snippets and combine them for classification. To encourage the model to discover diverse events, we apply dropout [19] throughout the model (and in particular on $\mathbf{s}$) to prevent the $\mathbf{v}^i$'s from co-adaptation. As we will see in the empirical analysis, our model automatically discovers events types that are semantically meaningful, and are highly correlated to the scene classes.

Table 1: *Class distributions with number of training samples and recording sessions.*

Activity	#10s seg	# sessions
Absence (nobody in room)	18860	42
Cooking	5124	13
Dishwashing	1424	10
Eating	2308	13
Other (no relevant activity)	2060	118
Social activity (visit, phone call)	4944	21
Vacuum cleaning	972	9
Watching TV	18648	9
Working (typing, mouse click, ...)	18644	33
Total	72984	268

3. Related work

Here we briefly describe a few previous approaches on DCASE 2018 Task 5, which we will use in the experiments. The baseline system provided by the organizer [20] was based on a 1-D CNN model applied to input Mel-spectrogram features extracted from the original 10-second clips. They treated each of the 4 channels of the audio clip as one independent data point during training, and this strategy was adopted by most of the teams. Among the top winning teams of the challenge, [21] similarly adopted a CNN-based architecture, except that they applied 2D convolutions along both the time and frequency dimension, which gave significant improvement over the baseline results. [22] applied heavy feature engineering and pre-processing techniques, such as blind dereverberation, blind source separation, and noise reduction, as their 'Front-End Modules', which can potentially be useful to our model as well. [23] applied sub-band convolutions in their architecture, and performed frame-wise prediction of scene label, with frame targets generated heuristically based on energy. [24] learned one shared network from two task—scene classification and regression, where the regression task is to predict pre-computed single-channel representations from multi-channel input data, and showed that the regression task helps improve classification performance.

4. Experimental results

We demonstrate the proposed method on the task 5 of DCASE 2018 challenge [25], an acoustic scene classification task derived from the SINS dataset [26]. In this dataset, each input audio clip is 10 seconds long, consisting of four acoustic channels with the sampling rate of 16 kHz.

We ignore the correlation between different channels, and treat the data from each channel as independent for training, as is done in previous work [22]; this yields a 4x augmentation of the training set size from 73K to 292K. During inference, for each audio clip, we obtain the predictions for each channel, and average the four prediction scores for the final classification.

4.1. Data augmentation

Proper data augmentation is important for our task, as the class distribution is very skewed, as shown in Table 1: the smallest class *Vacuum cleaning* contains only 972 training samples, compared to the class *Absence* which contains $18,860$ training samples. Previously, [28] augmented the data by randomly selecting two audio clips from the same class, evenly splitting each clip into five 2-second segments, and randomly selecting 5 out of the 10 segments, and concatenating them (in random or-

Table 2: *Configuration of the feature extraction network in proposed model. Note that all Conv layers below contain batch normalization [27] and ReLU activation.*

layer	kernel size	stride	# filters	data shape
Input				(64, 1250)
Conv	3x3	1x1	64	
Conv	3x3	1x1	64	
MaxPool	2x2	2x2		(32, 625, 64)
Conv	3x3	1x1	128	
Conv	3x3	1x1	128	
MaxPool	2x2	2x2		(16, 312, 128)
Conv	3x3	1x1	256	
Conv	3x3	1x1	256	
Conv	3x3	1x1	256	
MaxPool	2x2	2x2		(8, 156, 256)
Conv	3x3	1x1	512	
Conv	3x3	1x1	512	
Conv	3x3	1x1	512	
MaxPool	2x1	2x1		(4, 156, 512)
Conv	3x3	1x1	512	
Conv	3x3	1x1	512	
Conv	3x3	1x1	512	
MaxPool	2x1	2x1		(2, 156, 512)

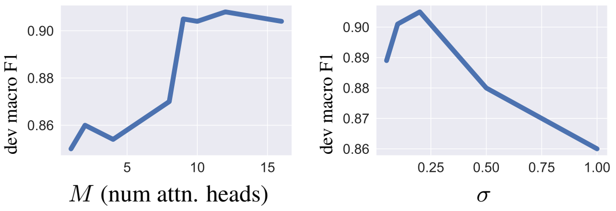

Figure 1: *Sensitivity analysis of hyperparameters* (M, σ)*. Left: dev performance for different* M*, with* $\sigma = 0.2$*. Right: dev performance for different* σ*, with* $M = 9$.

der) to form a new 10-second clip as augmented data for training.

We adopt a similar but simpler approach. We randomly select two clips within the same class, and from each clip, we cut out a continuous 5-second segment, whose starting time is sampled from $[0, 5]$ uniformly at random. The two 5-second segments are then concatenated to form a new 10-second clip for training. Augmentation is done on-the-fly during training, and model won't see the same augmented clip two times. We apply this augmentation strategy to the minority classes, namely *cooking*, *dishwashing*, *eating*, *other*, *social activity* and *vacuum cleaning*.

4.2. Model architecture

For input features, we extract 64D log-Mel features from the original single-channel audio data, with a window size of 16 ms and hop size of 8 ms, followed by per-utterance mean subtraction. This gives us a 64x1250 feature matrix per utterance.

Our feature extraction network f starts with VGGish ConvNets layers [29], the details of which are given in Table 2. The output of the last convolutional layer, with a receptive field of 64 ms in the time axis, is then fed to a bi-directional LSTM layer [9] to extract abstract features with rich temporal information. After that, we apply the multi-head attention module described in Section 2. This attention module outputs M (e.g., 9) fixed-dimension feature vectors, which are concatenated to form the final feature vector $\mathbf{s}$. The final classification network g consists of 2 hidden layers with 512 ReLU units [30] each, and a final softmax layer for 9-way classification.

Table 3: *Final performance (macro F1 score) of our method.*

Class	Baseline	[28]	Max Pool	Ours multi. attn.
Absence	0.877	0.937	0.896	0.927
Cooking	0.930	0.915	0.935	0.938
Dishwashing	0.772	0.865	0.829	0.866
Eating	0.812	0.870	0.849	0.880
Other	0.350	0.542	0.533	0.588
Social activity	0.966	0.979	0.977	0.979
Vacuum clean.	0.958	0.971	0.962	0.953
Watching TV	0.999	0.999	0.998	1.000
Working	0.814	0.887	0.822	0.884
Overall	0.831	0.884	0.867	**0.891**

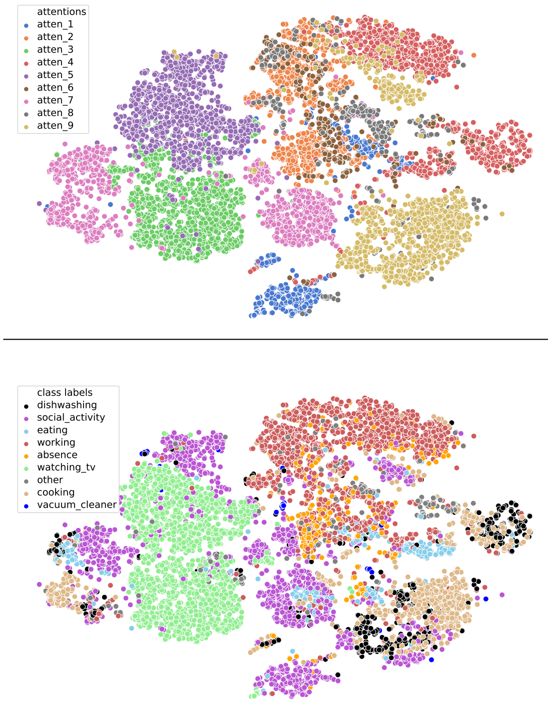

Figure 2: *2D t-SNE visualization of* $\mathbf{h}_t$*'s selected by the attention heads. Top plot is colored according to attention head, and bottom plot is colored according to utterance label.*

4.3. Model training and selection

We adopt the same strategy as the baseline method [25] to define an "epoch": we down-sample each class (by random sampling) to have same number of samples as the smallest class, going through these samples once constitutes one epoch, and we repeat the down-sampling process before each epoch. For training, we used the Adam optimizer [31] with minibatches of 200 utterances and an initial learning rate of 0.001. Furthermore, we reduce the learning rate by a factor of 0.5 every 7 epochs. We evaluate our model on the dev set every 5 epochs, and stop training when the dev set performance, measured by macro F1 score (which is also the final metric used by the challenge), does

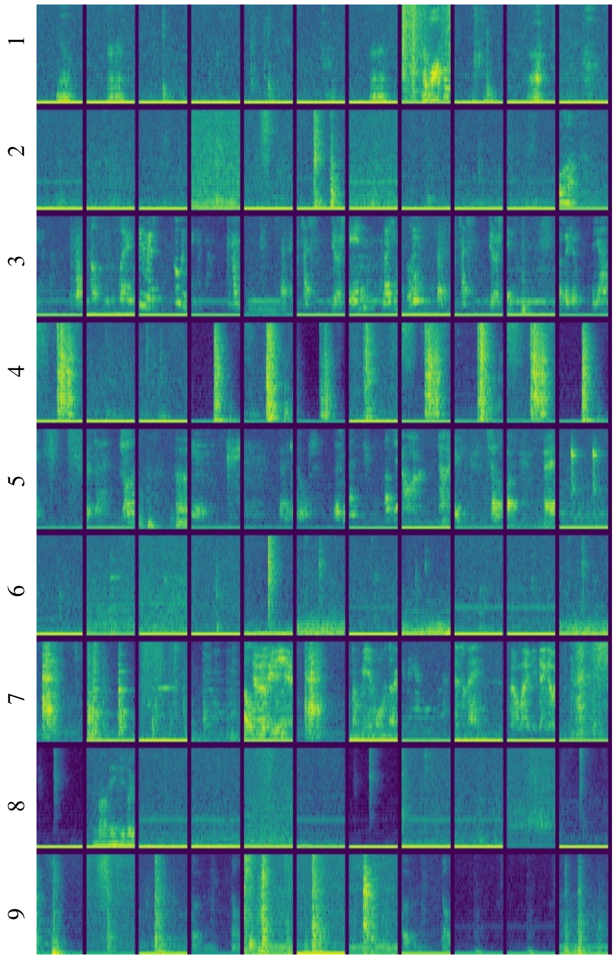

Figure 3: *Selection of* 1*-second segments selected by the attentions heads.*

not improve further.

We tune the two hyperparameters in our method—the number of attention heads M and the attention shape parameter σ—by grid search, based on dev set macro F1 score. In Fig. 1, we show how the dev performance changes as we vary one hyperparameter while fixing the other. From the top plot, we observe that for our task, the performance initially improves as we increase the number of attention heads M, and there is a significant gain when M reaches 9 (which coincidentally agrees with the number of classes), and the performance stabilizes after that. Therefore, we set $M = 9$ to balance performance and computational cost. The bottom plot shows the dev performance as we vary σ, and $\sigma = 0.2$ is chosen for the final model. Observe there exists a range of hyperparameters for which our model works similarly well.

4.4. Results

We show our final results on the evaluation set in Table 3. For comparison, we include also the results obtained by the baseline model, and the method from the winner of the challenge [28]; the performance of our method (5th column) is on par with the winner's solution. For ablation study, we also provide the performance of a variant of our method which, instead of using multi-head attentions, performs global max pooling on $f(\mathbf{X})$ in the time axis, for computing the utterance representation. This variant is denoted by "Max Pool" in Table 3 (4th column), whose performance significantly degrades from that of our final model. This demonstrates the effectiveness of the proposed multi-head attention mechanism.

5. Attention visualizations

In this section, we try to interpret the attention heads. Since the attention score a_t^i measures the level of relevance of frame feature $\mathbf{h}_t$ to event i, for each audio clip in the dev set, we select the $\mathbf{h}_t$ from the bi-directional LSTM outputs that is most aligned with each event i (i.e., $t = \arg\max_{t'} a_{t'}^i$), and visualize them with t-SNE [32] in 2D, as shown in Fig. 2. In the top plot, we color each point according to their attention head, whereas in the bottom plot, we visualize each point according to the class $\mathbf{h}_t$ comes from.

As we can see from the left plot, the $\mathbf{h}_t$'s show strong clustering associations with each attention heads, implying that the each attention head focuses more or less on a unique sound pattern (event). From the right plot, we discover correlations between feature representations and class labels. Notably, the pattern detected by attention 3 almost entirely belongs to *watching TV*. But in general, the correlation between attention heads and classes is not one-to-one. For example, head 5 may also contributes significantly to *watching TV*; on the other hand, some attention heads (such as 4 and 9) cover multiple closely related classes.

To understand the events each head attends to, for each $\mathbf{h}_t$ selected by head i, we find the corresponding time stamp on the original 10-second clip, and select a 1-second audio segment around the time stamp and listen to it. A sample of the log-Mel features for these 1-second segments are provided in Fig. 3. Each row refers to one attention head. We find each head attends to one or a few distinctive sound patterns. For example, attention heads 3 and 5 mainly detect human speech as well as media speech and phone conversations; these events are associated with *social activities* and *watching TV* in Fig. 2. Attention head 4 and 9, on the other hand, detect mostly percussive sounds like keyboard typing and mouse clicking in *working*, and silverware clanking and hitting sounds, shared by *cooking*, *eating*, and *dishing washing* in Fig. 2.

6. Conclusion

In this paper, we have proposed a multi-head attention model, which achieves competitive performance for acoustic scene analysis on DCASE 2018 competition dataset. The multi-head attention mechanism can discover meaningful representations of distinctive sound events and locate their appearances in time, given only class labels of the entire audio clip. Moreover, all parameters in our model can be trained jointly, in an end-to-end fashion. In future work, we may explore the proposed model with even more complex scenes and larger number of classes, and investigate different model architectures for ASC such as multi-level attention [33] and neural architecture search [34].

7. Acknowledgements

The authors would like to thank Harsha Sundar for helpful discussions.

8. References

[1] S. Ntalampiras, I. Potamitis, and N. Fakotakis, “On acoustic surveillance of hazardous situations,” in *ICASSP*, 2009.

[2] M. Lammers, R. Brainard, W. Au, T. Mooney, and K. Wong, “An ecological acoustic recorder (ear) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats,” *The Journal of the Acoustical Society of America*, 2008.

[3] J. Schroeder, S. Wabnik, P. Van Hengel, and S. Goetze, “Detection and classification of acoustic events for in-home care,” in *Ambient assisted living*. Springer, 2011.

[4] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *ICASSP*, 2017.

[5] T. Virtanen, A. Mesaros, T. Heittola, A. Diment, B. Elizalde, B. Raj, and E. Vincent, “Ieee aasp challenge on detection and classification of acoustic scenes and events,” 2017.

[6] A. Mesaros, T. Virtanen, T. Heittola, and F. C. et al, “Ieee aasp challenge on detection and classification of acoustic scenes and events,” 2018.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, 1997.

[10] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *NIPS*, 2015.

[11] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *ICASSP*, 2016.

[12] D. B., K. C., and Y. B., “Neural machine translation by jointly learning to align and translate,” in *ICLR*, 2015.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017.

[14] Y. Wang, J. Li, and F. Metze, “Comparing the max and noisy-or pooling functions in multiple instance learning for weakly supervised sequence learning tasks,” in *Interspeech*, 2018.

[15] B. McFee, J. Salamon, and J. P. Bello, “Adaptive pooling operators for weakly labeled sound event detection,” *CoRR*, vol. abs/1804.10070, 2018.

[16] Y. Wang, J. Li, and F. Metze, “A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling,” in *ICASSP*, 2019.

[17] W. Wang, C. Kao, and C. Wang, “A simple model for detection of rare sound events,” in *Interspeech*, 2018.

[18] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “DCASE 2017 challenge setup: Tasks, datasets and baseline system,” in *Proc. Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017.

[19] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, 2014.

[20] Q. Kong, I. Turab, X. Yong, W. Wang, and M. D. Plumbley, “DCASE 2018 challenge baseline with convolutional neural networks,” DCASE2018 Challenge, Tech. Rep., 2018.

[21] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, N. Greco, and R. Tachibana, “Domestic activities classification based on CNN using shuffling and mixing data augmentation,” DCASE2018 Challenge, Tech. Rep., 2018.

[22] R. Tanabe, T. Endo, Y. Nikaido, T. Ichige, P. Nguyen, Y. Kawaguchi, and K. Hamada, “Multichannel acoustic scene classification by blind dereverberation, blind source separation, data augmentation, and model ensembling,” *DCASE 2018 Challenge*, 2018.

[23] H. Liao, J. Huang, S. Lan, T. Lee, Y. Liu, and M. Bai, “DCASE 2018 task 5 challenge technical report: Sound event classification by a deep neural network with attention and minimum variance distortionless response enhancement,” DCASE2018 Challenge, Tech. Rep., 2018.

[24] K. Nakadai and D. R. Onishi, “Partially-shared convolutional neural network for classification of multi-channel recorded audio signals,” DCASE2018 Challenge, Tech. Rep., 2018.

[25] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, “DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics,” Tech. Rep., 2018.

[26] G. Dekkers, S. Lauwereins, B. Thoen, M. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, “The SINS database for detection of daily activities in a home environment using an acoustic sensor network,” in *DCASE*, 2017.

[27] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *ICML*, 2015.

[28] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, N. Greco, and R. Tachibana, “Domestic activities classification based on cnn using shuffling and mixing data augmentation,” *DCASE*, 2018.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv:1409.1556*, 2014.

[30] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *ICML*, 2010.

[31] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *ICLR*, 2014.

[32] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, 2008.

[33] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, “Multi-level attention model for weakly supervised audio classification,” in *DCASE2018 Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.

[34] J. Li, C. Liang, B. Zhang, Z. Wang, F. Xiang, and X. Chu, “Neural architecture search on acoustic scene classification,” Dec. 30 2019, arXiv:1912.12825 [cs.SD].