INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Deep Residual Network for Large-Scale Acoustic Scene Analysis

Logan Ford, Hao Tang, François Grondin, James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
{lhford,haotang,fgrondin,glass}@mit.edu

Abstract

Many of the recent advances in audio event detection, particularly on the AudioSet data set, have focused on improving performance using the released embeddings produced by a pre-trained model. In this work, we instead study the task of training a multi-label event classifier directly from the audio recordings of AudioSet. Using the audio recordings, not only are we able to reproduce results from prior work, we have also confirmed improvements of other proposed additions, such as an attention module. Moreover, by training the embedding network jointly with the additions, we achieve an mAP of 0.392 and an AUC of 0.971, surpassing the state of the art without transfer learning from a large data set. We also analyze the output activations of the network and find that the models are able to localize audio events when a finer time resolution is needed.

Index Terms: acoustic scene analysis, audio classification, audio event detection

1. Introduction

Audio classification is the task of detecting whether an event occurred given an audio clip. Examples could include recognizing car horns in driving footage or detecting glass breaking in a home security recording system. Audio classification was initially addressed as a single-label task [1, 2, 3, 4, 5, 6]. However, by definition, it is a multi-label classification task, because multiple audio events can occur in the same audio clip. There has been a lot of progress thanks to many publicly available datasets, such as ESC, those in the DCASE challenges, and AudioSet [7, 8, 9, 10]. Audio classification serves as the first step not only for understanding the environment but also for many downstream tasks, such as voice activity detection before speech recognition [11], noise detection before speech enhancement [12], and localizing speakers before speaker identification [13]. Any improvement in audio event detection can potentially help improve the downstream tasks.

Audio segmentation is a separate but similar task to audio classification. In classification, the goal is to simply recognize when a particular type of sound occurs somewhere in the input. In segmentation, the goal is to be able to detect exactly where in the sample each label occurred, more finely labeling the events within the audio scene over time. Audio classification is a weaker task in the sense that if there is a solver for audio segmentation, we can use that solver to solve audio event detection. On the other hand, while collecting fine-grained labeling for audio segmentation requires expert knowledge and is labor-intensive, collecting labels for audio classification does not require expert knowledge and can be easily crowdsourced.

In this work, we focus on deep convolutional neural networks (CNNs) for audio classification due to the success in many prior studies [14, 15, 16, 17, 18, 19]. CNNs perform especially well when there is access to a large amount of data. In particular, it has been shown in [20] that a CNN trained on 70 million audio clips on YouTube is able to generalize well to other audio clips collected in the wild, such as AudioSet [10].

A set of features extracted from a model in [20] has been released in the public domain, and a significant amount of work has been done based on these features. The study in [21] applies an attention module for each class over time after a few additional transformations on top of the released features. This approach is able to surpass the results on AudioSet in [20]. This work is extended by [22], applying a multi-level attention module. In particular, the output of attentions applied at different layers are concatenated before the final classification. This approach has the current state of the art AUC result on AudioSet, significantly surpassing their previous work. Most recently, this work has once again been extended by [23], where they applied an attention module in the hidden layers before a final classification layer, holding the current state of the art for mAP.

Though the released features certainly help make progress in this field, the results are not satisfying for the following reasons. First, we ignore if pre-training on 8 million audio clips is necessary to perform well on AudioSet. Second, we are unable to state that improvement from the additions, such as the attention module, would transfer without pre-training. Finally, having to work on released features without access to the actual working model hinders the possibility to update the base model based on the error signal from the additions, i.e., training the model end to end.

In this work, we explore these questions, by having a clean setting and clean comparisons for the task and considering some of the additions proposed in the past. Specifically, we find that training on AudioSet itself is sufficient to perform well on its evaluation set. Moreover, we do see performance improvement in the clean setting when the additions are used and when the models are trained end to end. We also explore several different architectures, and achieve a new state of the art without pre-training on 8 million audio clips or any other outside data.

Since audio classification is a weaker task than audio segmentation, it is natural to ask whether the trained network performs localization before giving the final classification result. We analyze the best performing network on a subset of AudioSet that contains fine-grained segmentations, and we find that this is indeed the case. At the penultimate layer just before applying attention in the network, we are able to extract the location at which the sound events occur. This further justifies the use of the weak labels collected for audio classification.

http://dx.doi.org/10.21437/Interspeech.2019-2731

2. Task Definitions

In this section, we formally define the task of audio classification. Let $\mathcal{X}$ be the space of input frames. For example, $\mathcal{X} = \mathbb{R}^{64}$ if the input feature is 64-dimensional log Mel spectrograms. Let $\mathcal{Y}$ be the space of labels, and in this case, $\mathcal{Y} = \{0, 1\}^K$, where K is the number of audio events that we are interested in detecting. The task of audio classification is to find a function that maps T input frames $x_1, \ldots, x_T$ to a vector $y \in \mathcal{Y}$ where $x_t \in \mathcal{X}$ for $t = 1, \ldots, T$. For audio segmentation, the goal is to find a function that maps T input frames $x_1, \ldots, x_T$ to a sequence $(c_1, s_1, e_1), \ldots, (c_m, s_m, e_m)$ of m events where s_i and e_i are the start and end time and $c_i \in \{1, \ldots, K\}$ is the class of the i-th event for $i = 1, \ldots, m$. Note that in general the number of frames T varies from clip to clip, and the number of events m can be zero. It is also easy to see that if a vector y has $y_{c_i} = 1$ for $i = 1, \ldots, m$ and zero everywhere else, then this vector y is a correct audio classification so long as the audio segmentation is correct.

3. Model Architectures

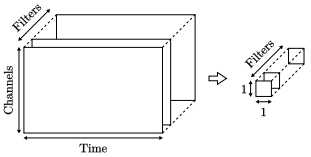
In this section, we describe the proposed and evaluated ResNet variants. ResNet [24] was the first model to introduce skip-connections, additional layers that pass values around future layers, allowing blocks of layers to easily learn the identity function and avoid transformation of features. Skip-connections also enable improved error propagation and significantly help against the vanishing gradient problem. This allows for the extending of a network with many more layers with little worry of degradation in performance. The ResNet architecture has been used in audio tasks such as speaker spoof detection [25] and unsupervised audio representation learning [26].

We start with the ResNet50 audio variant proposed by [20], termed here **ResNet A**. It consists of 50 layers with a skip connection almost every 3 layers, with an average pooling before prediction. Following [20], we divide each 10s audio clip into 960ms independent segments, feed each segment into the network to make independent predictions, then average the individual predictions over all segments.

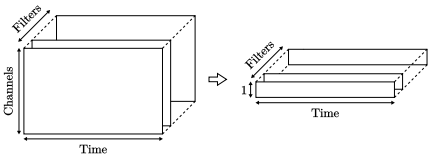
The first variation we explored eliminated the segmentation step used by ResNet A, and processed the entire 10s audio clip, advancing one 10ms frame at a time. As shown in Figure 1, this effectively produced a 3D tensor consisting of time (T), Channels (filterbanks), and CNN filters (F). The **ResNet B** model performed Global pooling over this tensor at the penultimate layer, producing a single F-dimensional vector, which was passed to the final classification layer. Global pooling thus averages over time and frequency channels.

In order to be able to attend over time, we then considered a different kind of pooling that we call Channel pooling, which average pooled only over the Channel dimension. Channel pooling is shown in Figure 1 (b). Model **ResNet B + att** incorporated an attention mechanism with Channel pooling.

Finally, we also considered two other ResNet models using different residual architectures. **ResNet C + att** and **ResNet D + att** are the same as **ResNet B + att** but are based on ResNet34 and ResNet101, respectively. Table 1 describes these model architectures in detail.



(a) *Global pooling*



(b) *Channel pooling*

Figure 1: *Global vs Channel pooling. In global pooling, all elements in the time and channel dimensions are averaged to generate a* $1 \times 1 \times F$ *tensor, with* F *being the number of filters. Alternatively, channel pooling averages over channels to generate a* $1 \times T \times F$ *tensor, where* T *is the number of frames, which allows for using an attention mechanism over time.*

4. Experiments

4.1. Dataset

Our work focused on utilizing AudioSet, a collection of over 2 million 10-second clips of YouTube videos released by Google, weakly labeled with the sounds that the clip contains from a set of 527 labels. Weakly labeled, as opposed to strongly labeled, means that labels are given to a clip with no indication of where in the clip the associated sound occurred. AudioSet is also a multi-label dataset so every clip can, and most often does, have multiple labels associated with it, with an average of 5 labels per sample. The dataset is split into three groups: `balanced_train`, `unbalanced_train`, and evaluation. The `balanced_train` dataset is a set of 22,000 examples, where each label has 49 samples, while the `unbalanced_train` set contains the rest of the complete training dataset. The evaluation set consists of 22,000 examples. AudioSet indexes video IDs, timestamps, and labels for each video segment from YouTube. It also provides bottleneck features, which consist of 128-D vectors for each second of audio, and were obtained using a VGG-inspired model trained on an early version of the YouTube-8M dataset.

We extracted the dataset from YouTube, but due the constant change in video availability (videos being removed, taken down, etc.) there is a natural shrinkage (about 5%) from the original dataset. This noted, we do draw fair comparisons between the previous state-of-the-art architecture and our models by evaluating on the same subset of the evaluation dataset.

4.2. Evaluation

We train our model to be able to predict the classes that occur at some-point in an approximately 10 second long audio clip. We have our model predict each label independently, as multiple labels can occur in a single sample. Our model predicts softly, returning some value between 0 and 1. It is also worth noting that while the labels of AudioSet form an ontology, each label is still predicted independently, partly due to the fact that the appearance of a ground truth label does not imply the presence of parent and ancestor labels.

For evaluation, we measure how our model performs on the

Table 1: *Tested ResNet Architectures. Note Truncated ResNet is the ResNet architecture after applying changes described by [20], and removing all layers after the final convolution layer.*

Model	ResNet A	ResNet B	ResNet B + att	ResNet C + att	ResNet D + att
Input	11x960ms	10s	10s	10s	10s
Base Model	Truncated ResNet50			Truncated ResNet34	Truncated ResNet101
Pooling	Global Pool	Global Pool	Channel Pool	Channel Pool	Channel Pool
Extension	1FC + Time Averaging	1FC	1FC + Attention Module	1FC + Attention Module	1FC + Attention Module

AudioSet evaluation set by three metrics: 1) mean Average Precision (mAP) of all the classes, which is an approximation of the area under a class's precision-recall curve, 2) average area under the curve (AUC) of the receiver operating characteristic (ROC) curve for each class, and 3) sensitivity index (d-prime), which is deterministically calculated from AUC.

4.3. Training

Each model was trained on the whole subset of the AudioSet training set that was still available on YouTube at the time of extraction. Our training subset consists of 1,953,082 samples of the total 2,063,949, a 5.3% loss from the original dataset. We trained each model for up to 50 epochs, with most all models peaking in performance by the 40th epoch. All networks were trained with the Pytorch framework [27], using the Adam optimizer [28], with a learning rate of 0.0001, weight decay of $5\mathrm{e}{-7}$, and beta values of 0.95 and 0.999.

4.4. Results

Our evaluation subset consists of 19,185 samples; 5.9% fewer than the original 20,371. To ensure we could compare to the state-of-the-art method, we re-trained the Multi-level attention model by Yu et al.[22] on the released features and evaluated the final model-averaged results on our evaluation subset. As the results were slightly worse overall (mAP: 0.3586, AUC: 0.9678), we believe that the published numbers are a fair comparison.

As seen in Table 2, there's an interesting pattern in our transition from ResNet A, to ResNet B, to ResNet B + att. Performance worsened when feeding the whole audio clip into the ResNet variant model (B). The addition of an attention module greatly improved our performance, leading to our checkpoint averaged result achieving state-of-the-art performance (ResNet B + att (Avg)), and even our peak individual model (ResNet B + att) still outperforms previous state-of-the-art results. It is also clear in the progression from our shallower to deeper models (C, B, D) that there is clear performance gain moving from the ResNet34 to the ResNet50 based model, but insignificant gains moving deeper from ResNet50 to ResNet101 based models.

Table 2: *Model Performance on AudioSet Evaluation set. Note that (Avg) indicates checkpoint averaged results, where outputs of all 50 epoch versions are averaged and evaluated.*

Model	mAP	AUC	d-prime
Baseline[20]	0.314	0.959	2.452
Yu[22]	0.360	0.970	2.660
Kong[23]	0.369	0.969	2.640
ResNet A	0.347	0.966	2.582
ResNet B	0.329	0.966	2.584
ResNet B + att	0.379	0.970	2.657
ResNet B + att (Avg)	**0.392**	**0.971**	**2.682**
ResNet C + att	0.360	0.966	2.587
ResNet D + att	0.380	0.970	2.655

Table 3: *Segmentation Performance (F1 Macro %)*

Model	Test	Eval
DCASE Baseline	14.06	-
JiaKai[29]	25.9	32.4
0.5 Thresholds	9.83	6.70
Swept Thresholds (ST)	12.41	8.70
median filtering and ST	17.87	11.50

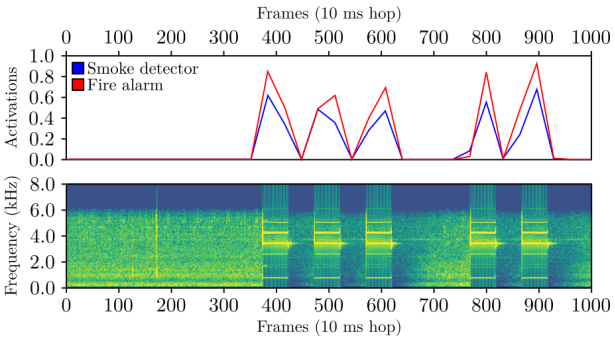
5. Segmentation

An auxiliary task that our strongest model is able to perform, thanks to the addition of the attention module, is audio event segmentation. This means that although our model was trained on weakly-labeled data, it is able to predict strongly.

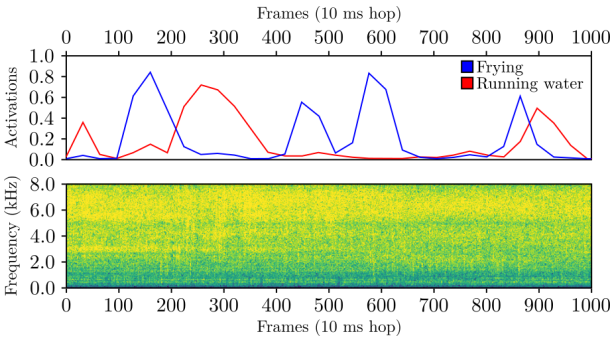
To evaluate our model, we made use of data from task 4 of DCASE 2018, "Sound event detection in domestic environments" [14]. In particular, we used the strongly labeled portions of the data which were the test set released at the start of the challenge, and the final evaluation set. To perform segmentation with our model architecture, we took the output right before attention is applied over time and applied a class-dependent threshold to determine activation. We show results for threshold of 0.5 for all classes, as well as fine-tuned thresholds which were found by sweeping over values and evaluating on the test set by each class's F1 score in Table 3. We then also swept over kernel sizes for class-specific median filters, allowing for smoothing over activation values before thresholds are applied.

Performance on this task is evaluated with the macro-averaged F1 score. This means finding the F1 score per label and taking the unweighted mean of these values. To compute an F1 score, there needs to be a way to determine a hit or miss for each strong label. A labeling is determined to be a hit if both the starting and end times are within 200ms of the ground-truth, and that the event duration is no less than 80% of the true event duration [30]. A labeling that does not fit this criteria is a false-positive, and any ground-truth label that does not have a corresponding "hit" labeling is considered a false-negative.

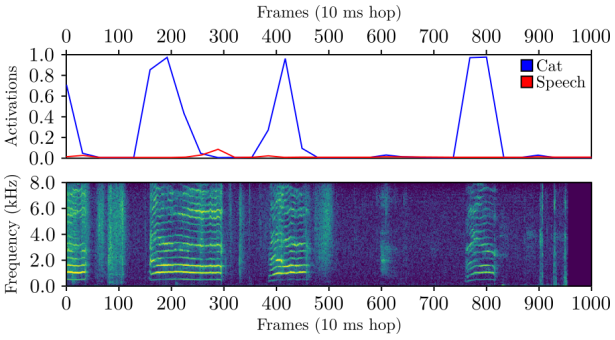
There are a few things to note in reviewing these results. For one, our model has a time resolution of about 300ms but



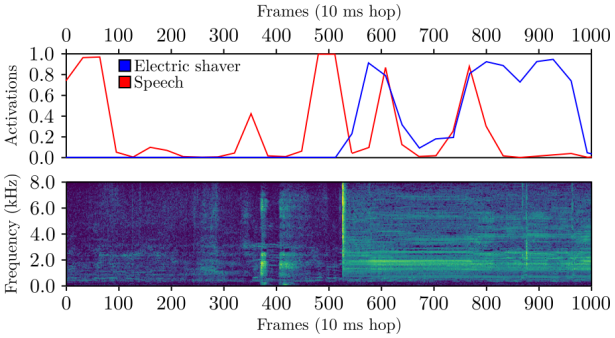
(a) *Clear detection of a fire alarm*



(b) *Frying, a long-duration event, is not consistently being detected despite its presence through the entire audio sample*



(c) *Clear detection of cat noises*



(d) *Shaver detection is not consistent throughout the event, and speech is being sporadically triggered by background mumbling*

Figure 2: *Spectrograms and activation levels for different sound events. The activation plots show only the top-2 classes for clarity.*

Table 4: *Segmentation Performance by Class (F1 %)*

Class	Our model	Baseline
Alarm/bell/ringing	**9.8**	3.9
Blender	3.4	**15.4**
Cat	**40.0**	0.0
Dishes	**20.5**	0.0
Dog	**21.1**	0.0
Electric Shaver/Toothbrush	24.5	**32.4**
Frying	12.5	**31.0**
Running Water	1.9	**11.4**
Speech	**27.3**	0.0
Vacuum Cleaner	17.9	**46.5**

the evaluation is done with a 200ms collar. This means that there is naturally going to be some loss in performance purely due to this discretization. When the time collar is increased to 400ms, Macro F1 performance increases to 31.25% on the test set and 22.26% on the eval set, indicating that the model is detecting many of the events but with lesser precision. In the release of the baseline, they claim that their model doesn't seem to be really learning how to perform segmentation as it's only succeeding with long-duration class types, which in many clips is essentially equivalent to a clip-level classification. In contrast, our model performs particularly well with short-duration events, and worse with longer-duration events (Table 4). This shows that our model is in fact learning how to perform segmentation given the success with short-duration event types. A possible reason for failure in long-duration events is that the model found some features in time to be more strongly indicative of a classification result, leading to inconsistent detection throughout the event's entirety. Figure 2 shows both positive and negative examples of our segmentation results. Finally, we can see that while our model is able to outperform the DCASE baseline, the top performing models built specifically for segmentation achieve better results [29].

6. Conclusion

This paper presents a model that outperforms previous state-of-the-art work on the AudioSet dataset labeling task, surpassing results from models built upon the released bottleneck features, achieving an mAP of 0.392 and AUC of 0.971. This indicates that there is promise in further exploring model types that learn directly from log-Mel features of the audio samples. We also validated that the addition of an attention module helps significantly in improving the performance of CNN architectures for Audio Classification. The attention module has also shown to perform segmentation in order to achieve a final classification, and can be used to extract strong labels for acoustic events.

Work here has indicated that there is a potential greater need for capturing long-duration acoustic characteristics. Model architectures employing some form of recurrent structure could be particularly useful in more accurately performing segmentation, and increasing confidence of detecting a particular sound given surrounding key features.

7. Acknowledgements

This work was supported in part by Signify.

8. References

[1] A. B. Nielsen, L. K. Hansen, and U. Kjems, "Pitch based sound classification," in *Proc. IEEE ICASSP*, vol. 3, 2006, pp. 788–791.

[2] Y. Sasaki, M. Kaneyoshi, S. Kagami, H. Mizoguchi, and T. Enomoto, "Daily sound recognition using pitch-cluster-maps for mobile robot audition," in *Proc. IEEE/RSJ IROS*, 2009, pp. 2724–2729.

[3] F. Grondin and F. Michaud, "Robust speech/non-speech discrimination based on pitch estimation for mobile robots," in *Proc. IEEE ICRA*, 2016, pp. 1650–1655.

[4] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," in *Proc. ACM Multimedia*, 2001, pp. 203–211.

[5] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Speech & Audio Process.*, vol. 10, no. 7, pp. 504–516, 2002.

[6] S. Dutta and A. Ghosal, "A hierarchical approach for silence/speech/music classification," in *Proc. IEEE ICPCSI*, 2017, pp. 3001–3005.

[7] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. EUSIPCO*, 2016, pp. 1128–1132.

[8] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. ACM Multimedia*, 2015, pp. 1015–1018.

[9] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM TASLP*, pp. 379–393, 2016.

[10] J. F. Gemmeke, D. P. 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 *IEEE ICASSP*, 2017, pp. 776–780.

[11] N. Cho and E.-K. Kim, "Enhanced voice activity detection using acoustic event detection and classification," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 196–202, 2011.

[12] S. Ravindran and D. V. Anderson, "Audio classification and scene recognition and for hearing aids," in *Proc. IEEE ISCAS*, 2005, pp. 860–863.

[13] S.-C. Liu, J. Bi, Z.-Q. Jia, R. Chen, J. Chen, and M.-M. Zhou, "Automatic audio classification and speaker identification for video content analysis," in *Proc. IEEE/ACIS SNPD*, vol. 2, 2007, pp. 91–96.

[14] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection," in *arXiv:1807.10501*, 2018.

[15] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. IEEE MLSP*, 2015, pp. 1–6.

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

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

[18] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE ICASSP*, 2015, pp. 559–563.

[19] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of lstm and cnn," in *Proc. DCASE*, 2016, pp. 11–15.

[20] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE ICASSP*, 2017, pp. 131–135.

[21] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Audio set classification with attention model: a probabilistic perspective," in *Proc. IEEE ICASSP*, 2018, pp. 316–320.

[22] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," in *arXiv:1803.02353*, 2018.

[23] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," in *arXiv:1903.00765*, 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[25] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatical spoofing detection," in *Interspeech*, 2017.

[26] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *IEEE ICASSP*, 2018.

[27] N. Ketkar, "Introduction to pytorch," in *Deep learning with python*. Springer, 2017, pp. 195–208.

[28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[29] L. JiaKai, "Mean teacher convolution system for DCASE 2018 task 4," in *Proc. DCASE*, 2018.

[30] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.