ENSEMBLE OF PRUNED LOW-COMPLEXITY MODELS FOR ACOUSTIC SCENE CLASSIFICATION

Kenneth Ooi[*], *Santi Peksi*[*], *Woon-Seng Gan*[*]

Nanyang Technological University
School of Electrical and Electronic Engineering
50 Nanyang Ave, Singapore 639798
{ wooi002, speksi, ewsgan }@ntu.edu.sg

ABSTRACT

For the DCASE 2020 Challenge, the focus of Task 1B is to develop low-complexity models for classification of 3 different types of acoustic scenes, which have potential applications in resource-scarce edge devices deployed in a large-scale acoustic network. In this paper, we present the training methodology for our submissions for the challenge, with the best-performing system consisting of an ensemble of VGGNet- and InceptionNet-based lightweight classification models. The subsystems in the ensemble classifier were pruned by setting low-magnitude weights periodically to zero with a polynomial decay schedule to achieve an 80% reduction in individual subsystem size. The resultant ensemble classifier outperformed the baseline model on the validation set over 10 runs and had 119758 non-zero parameters taking up 468KB of memory. This shows the efficacy of the pruning technique used. We also performed experiments to compare the performance of various data augmentation schemes, input feature representations, and model architectures in our training methodology. No external data was used, and source code for the submission can be found at https://github.com/kenowr/DCASE-2020-Task-1B.

Index Terms — Acoustic scene classification, weight pruning, ensemble classifier, VGGNet, InceptionNet

1. INTRODUCTION

Acoustic scene classification has been one of the mainstays of the DCASE Challenge. It aims to identify the environment in which an acoustic recording was made given the raw audio data itself. Prior to the DCASE 2020 Challenge, the focus of this task has been on the development of models with high classification accuracy. However, there is a well-known tradeoff between classification accuracy and model complexity, in that increasingly complex models are required to obtain higher classification accuracies. Hence, the focus of Task 1B has shifted to reflect this, by requiring models to achieve as high a classification accuracy as possible within a model size of 500 kilobytes (KB).

The main approaches to acoustic scene classification in the literature can be broken down into three main types: (i) data-driven approaches looking to modify or augment the given dataset, (ii) representation-driven approaches looking to transform the given raw audio data to a different, possibly more salient form, and (iii) model-driven approaches looking to find modules and architectures that best replicate the desired output for a given input. A brief overview of these techniques is as follows.

For data-driven approaches, other than the usage of external data, mixup augmentation [1]–[3] has been popular as a computationally cheap way to augment a dataset. In a similar fashion, Takahashi et al. proposed a method called Equalized Mixture Data Augmentation which creates new training samples from linear combinations of parametrically equalized versions of the original samples [4]. Furthermore, Chen et al. used a convolutional variational autoencoder (CVAE)/generative adversarial network (GAN) system in the DCASE 2019 Challenge, which makes use of a separate neural network that generates new training samples, but is more computationally heavy [5].

For representation-driven approaches, log-mel spectrograms and mel-frequency cepstral coefficients of the raw audio data have commonly been used as input features to acoustic scene classification models. Alternatives to these features include mel-frequency discrete wavelet coefficients and constant-Q cepstral coefficients [6], a combination of chroma, spectral contrast, and tonnetz features [7], and separation into harmonic and percussive components [2]. In addition, several teams have made use of the binaural nature of the recordings to devise useful representations, such as through primary ambient extraction to generate 4-channel spectrograms [3], as well as generalized cross-correlation-phase-transform (GCC-PHAT) and interaural time difference (ITD) features [8].

For model-driven approaches, 2-dimensional (2D) convolutional neural network (CNN) classifiers have often been utilized in conjunction with spectrogram representations as input, given that spectrograms can be identified as images and that 2D CNNs have enjoyed much success in image processing tasks. Models exploiting the time-domain nature of the raw signals have also been used, such as 1D CNN-based classifiers [9], [10] and AclNet [11]. Moreover, some authors have also modified existing network architectures to better fit the acoustic domain. For example, McDonnell et al. used residual networks with parallel but separate pathways for high and low frequency components [12], Su et al. modified an Xception network to allow predictions with multi-scale features from outputs at different depths [7], and Koutini et al. modified ResNet and DenseNet to incorporate receptive-field regularization and frequency-awareness [13]. Other approaches include the application of Dempster-Shafer evidence theory to aggregate subsystem outputs into an ensemble classifier [14], as well as the usage of a domain adaptation network to cope with potential device mismatch problems [13]. Lastly, knowledge distillation also reduces model complexity,

[*]Supported by the Singapore Ministry of Education Academic Research Fund Tier-2, under research grant MOE2017-T2-2-060.

because a larger teacher model is used to train a smaller student model to mimic the teacher's outputs [15].

For our submission, we focused on the use of pruning low-magnitude weights to reduce model complexity. Initially proposed by Lecun et al. [16], pruning can potentially ameliorate overfitting problems with complex models while reducing the parameter count. Hence, we used relatively straightforward architectures and data preprocessing methods to observe their effects on classification accuracy.

2. DATA PREPROCESSING

For our submission, we used the TAU Urban Acoustic Scenes 2020 3Class dataset [17], [18], which consists of 10-second long binaural recordings captured at a 48 kHz sampling frequency [19]. There is a 70-30 split between the training set and validation set, which we respectively used to train and evaluate our models. All recordings are classified into 10 fine-grained classes, which are in turn classified into the 3 coarse-grained classes "indoor", "outdoor", and "transportation" for Task 1B.

2.1. Feature Extraction

We used log-mel spectrograms as the features to train all the models in our submission. The binaural recordings were first converted to mono recordings by taking the point-wise mean of sample values across channels. The log-mel spectrograms were then generated from the short-time Fourier transform (STFT) of the mono recordings using a Hann window of length 2048 with 50% overlap between windows and 48 mel bands with a minimum and maximum frequency of 0Hz and 24kHz, respectively.

The choice of the number of mel bands and STFT window length (with constant 50% overlap) was made as a result of a preliminary grid search over the sets {32, 48, 64} and {1024, 2048, 4096}, respectively. The search was conducted with the same model architecture (Model 1, as described in Section 3.1) for 200 epochs and without pruning. We chose the parameters with the best performance, which were indeed 48 mel bands and an STFT window length of 2048, as shown in Table 1.

2.2. Data Augmentation

We used a modified version of random block mixing [20] to augment the dataset. Each augmented track consists of ten 1-second long segments from different recordings in the original dataset that have been concatenated. The 1-second long segments for each augmented track were chosen at random points of random recordings belonging to the same coarse-grained class, but from as many different cities as possible to maximize variation in the augmented data. Hence, each augmented track has the same label as the original segments that comprise it. An example of an augmented track can be seen in Figure 1.

Before deciding on our data augmentation scheme, we also explored the effect of concatenating different numbers of segments (2, 4, 5, 10 segments of length 5, 2.5, 2, 1 seconds, respectively), as well as whether the random segments were from the same coarse-grained or fine-grained class. We used the same model architecture (Model 1) and pruning schedule (described in Section 4) for this experiment, and Table 2 shows the results.

Table 1: Unpruned Model 1 macro-averaged accuracy (mean ± standard deviation (SD)) over 10 runs with different log-mel spectrogram parameters

# mel bands	STFT window length		
	1024	2048	4096
32	0.8789 ± 0.0070	0.8738 ± 0.0078	0.8693 ± 0.0085
48	0.8776 ± 0.0057	0.8858 ± 0.0052	0.8797 ± 0.0097
64	0.8846 ± 0.0056	0.8848 ± 0.0052	0.8785 ± 0.0056

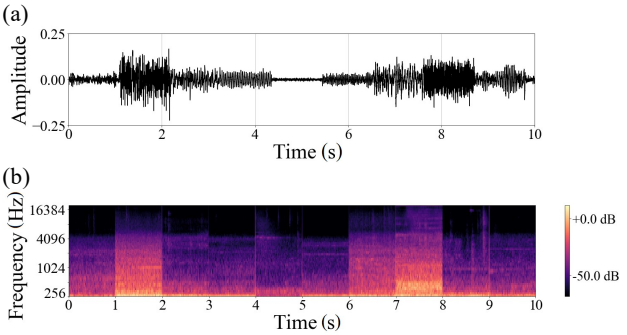

Figure 1: Example of augmented track with label "transportation" as a (a) time-domain signal and (b) log-mel spectrogram.

Table 2: Pruned Model 1 macro-averaged accuracy (mean ± SD) over 10 runs with different data augmentation schemes.

# segments	Type of class labels used for mixing	
	Fine-grained	Coarse-grained
2	0.8852 ± 0.0068	0.8887 ± 0.0060
4	0.8839 ± 0.0086	0.8895 ± 0.0048
5	0.8870 ± 0.0067	0.8821 ± 0.0057
10	0.8852 ± 0.0079	0.8840 ± 0.0069

There is almost no difference in mean macro-averaged accuracy among the different schemes. This was confirmed by Friedman test with two factors: the number of segments and type of class labels used for mixing. The p-values for both factors were respectively 0.3023 and 0.5453, indicating that both factors had no significant effect (at a 0.05 significance level) on model performance. Hence, we chose to use ten 1-second segments with identical coarse-grained labels to maximize the variance of the augmented data samples over time.

3. NETWORK ARCHITECTURE

The networks that we used for our submission to Task 1B were variants of VGGNet [21] and InceptionNet [22] using fewer filters per layer and fewer layers. Initially designed for image recognition tasks, they have also been adapted successfully for tasks in the audio domain in previous studies [5], [23]–[25].

3.1. Subsystem Architecture

VGGNet and InceptionNet respectively use stacks of smaller VGG(k) and Inception(k) modules, where k denotes the number of filters used for the convolutional layers in the modules. The structure of these modules is shown in Figure 2. We denote the

variants of VGGNet and InceptionNet used in our submission as V(a,b,c,d) and I(p,q,r), where (a,b,c,d) and (p,q,r) denote the sequence of numbers of VGG(k) and Inception(k) modules respectively present in their architectures. The networks had increasing numbers of filters in later layers, and are shown in full in Figure 3 and Figure 4. As advised in [22], to improve prediction accuracy, we did not perform batch normalization for the Inception(k) modules for the I(p,q,r) architecture, and used Inception(k) modules only at the latter layers of the network with regular convolutional layers at the beginning. All parameters in the subsystems used were in 32-bit floating point representation. We used (a,b,c,d) = (2,2,3,3) and (p,q,r) = (2,1,2) for the networks in our submission to keep the networks relatively shallow. We also explored the effects of altering the network depth after the challenge deadline, and present these results in Section 5.

3.2. Ensemble Classifier

In addition to the two basic networks described in Section 3.1, we also combined five VGGNet-based models (Figure 3) and one InceptionNet-based model (Figure 4), trained independently with different randomly-initialized weights on the same dataset, as subsystems for an ensemble classifier. The mean of the class probabilities from each subsystem was taken to be the output of the final ensemble classifier.

3.3. Submitted Models

The four models that we submitted are made up of different combinations of the network architectures described in Section 3.1, and are specifically described as follows.

- Model 1: V(0,0,1,0) trained on non-augmented data.
- Model 2: I(2,1,2) trained on non-augmented data.
- Model 3: Ensemble classifier (five V(0,0,1,0) models + one I(2,1,2) model) trained on non-augmented data.
- Model 4: Ensemble classifier (five V(0,0,1,0) models + one I(2,1,2) model) trained on augmented data.

4. TRAINING METHODOLOGY

Each model (or subsystem) in our submission was trained for 400 epochs with a batch size of 128 samples. An L2 kernel regularizer (regularization factor 0.001) was applied to all 2D convolutional and dense layers. We used the Adam optimizer with a learning rate of 0.0001 to train every model (or subsystem) by minimizing the regularized categorical cross-entropy loss between the predictions and ground-truth labels.

In addition, we adopted a pruning schedule during the training phase similar to that proposed by Zhu and Gupta in [26]. The pruning schedule has a polynomial decay as shown in Equation (1). We denote s_i as the initial sparsity and s_f as the final sparsity (proportion of model parameters set permanently to zero at the start and end of the pruning schedule, respectively). Let n be the number of times pruning occurs, t_0 be the first epoch when pruning occurs, and Δt be the number of epochs between each time pruning occurs, then we have

$$s_k = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3 \quad (1)$$

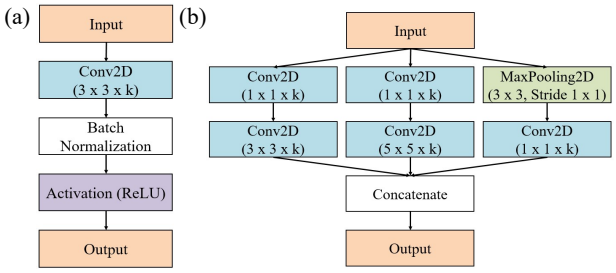

Figure 2: Architecture of (a) VGG(k) module and (b) Inception(k) module.

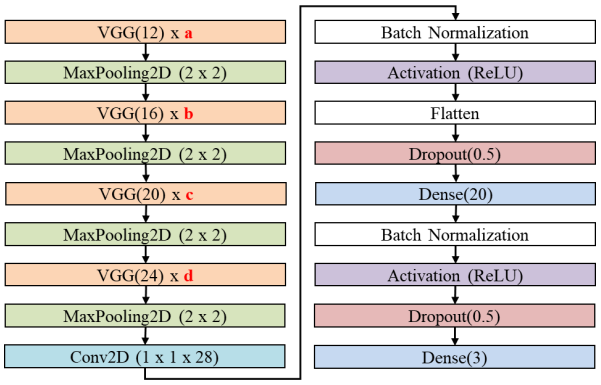

Figure 3: VGGNet-based network V(a,b,c,d) architecture. The parameter values a, b, c, and d are shown in red text.

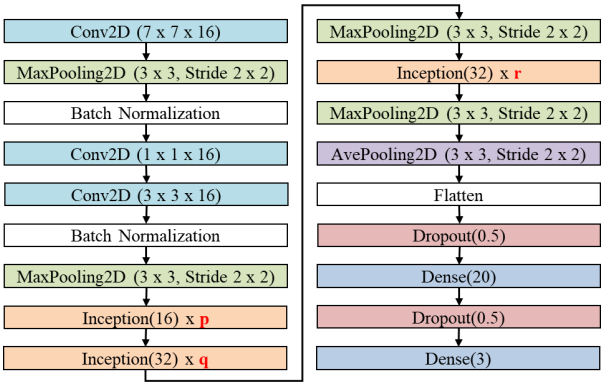

Figure 4: InceptionNet-based network I(p,q,r) architecture. The parameter values p, q, and r are shown in red text.

for all k in $\{t, t+1, \ldots, t+\Delta t\}$ and t in $\{t_0, t_0+\Delta t, \ldots, t_0+n\Delta t\}$. In our submissions, we used $s_i = 0.1$, $s_f = 0.8$, $n = 20$, $t_0 = 100$, and $\Delta t = 10$.

5. RESULTS AND DISCUSSION

We used the trained and pruned models to make predictions on the validation set, and compared them with the provided ground-truth labels to determine their micro-averaged and macro- averaged accuracies. For reference, we also trained unpruned

Table 3: Summary of submitted model performance (mean ± SD) over 10 runs on validation set. The baseline model performance (B), as reported by the challenge organizers, is also presented here for comparison. All models were trained over 400 epochs, but pruned models followed the schedule in Section 4. Model sizes were calculated based on number of non-zero parameters.

Model	Unpruned				Pruned			
	Accuracy (micro)	Accuracy (macro)	# non-zero parameters	Model size (KB)	Accuracy (micro)	Accuracy (macro)	# non-zero parameters	Model size (KB)
B	---	0.873 ± 0.007	115219	450.1	---	---	---	---
1	0.8896 ± 0.0052	0.8890 ± 0.0058	80839	315.8	0.8798 ± 0.0088	0.8797 ± 0.0080	17115	66.9
2	0.8734 ± 0.0044	0.8728 ± 0.0040	167571	654.6	0.8803 ± 0.0070	0.8797 ± 0.0069	34181	133.5
3	0.9112 ± 0.0040	0.9108 ± 0.0041	571768	2233.6	0.9060 ± 0.0020	0.9066 ± 0.0020	119758	467.8
4	0.9132 ± 0.0024	0.9130 ± 0.0024	571768	2233.6	0.9086 ± 0.0019	0.9090 ± 0.0020	119758	467.8

Table 4: Summary of performance unpruned model architectures of different depths (mean ± SD) over 10 runs on validation set.

Metrics	VGGNet-based models V(a,b,c,d)			InceptionNet-based models I(p,q,r)		
	V(0,0,1,1)	V(2,2,3,3)	V(4,4,5,5)	I(1,0,1)	I(2,1,2)	I(3,2,3)
Accuracy (micro)	0.8616 ± 0.0014	0.8920 ± 0.0045	0.8797 ± 0.0044	0.8891 ± 0.0018	0.8758 ± 0.0048	0.8801 ± 0.0043
Accuracy (macro)	0.8624 ± 0.0019	0.8921 ± 0.0040	0.8794 ± 0.0036	0.8890 ± 0.0016	0.8743 ± 0.0036	0.8789 ± 0.0043
# NZ parameters	54415	80839	106327	68099	167571	267042
Model size (KB)	212.6	315.8	415.3	266.0	654.6	1043.1

versions of the four models described in Section 3.3 for 400 epochs, and present a summary of the performance of both the unpruned (not submitted) and pruned (submitted) models in Table 3. All models in our submission exceeded the mean baseline model macro-averaged accuracy of 0.873. They were also within the size limit of 500KB. These results show that the combination of pruning, shallower models, and modified block mixing could improve classification accuracy for acoustic scene classification tasks.

With the pruning schedule described in Section 4, we can also see that all models in our submission had a five-fold reduction in number of non-zero (NZ) parameters and model size as compared to the unpruned models. In addition, the pruned ensemble classifiers (Models 3 and 4) performed markedly better than the pruned single models (Models 1 and 2), with an approximate 2% increase in both mean micro-averaged and macro-averaged accuracy over the single models.

Pruning also led to a drop in performance of all models except Model 2 by about 0.5-1% in both micro- and macro-averaged accuracy. In contrast, both micro- and macro-averaged accuracies of Model 2 *increased* with pruning. To quantify the significance of these differences, we performed two-sided Wilcoxon rank-sum tests between the results of the pruned and unpruned versions of each model. The p-values were 0.0210, 0.0140, 0.0090, and 0.0001 for the micro-averaged accuracies of Models 1, 2, 3, and 4, respectively, and 0.0155, 0.0173, 0.0538, and 0.0025 for the macro-averaged accuracies of Models 1, 2, 3, and 4, respectively.

The significance (at a 0.05 significance level) of the differences in performances of Models 1, 3, and 4 are as expected due to the decrease in number of parameters available to fit the data after pruning. However, the significance of the difference in performance of Model 2 hints at possible overfitting in the unpruned version, and that pruning could have precluded this issue precisely through parameter reduction. Since Models 1 and 2 are respectively VGGNet- and InceptionNet-based, the results could also indicate that different network architectures are amenable to pruning at different extents, but further investigation would be necessary to validate this hypothesis.

Lastly, Table 4 summaries the single unpruned model performances at different depths due to the choice of (a,b,c,d) and (p,q,r). The VGGNet-based and InceptionNet-based models with the best classification accuracy (both macro-averaged and micro-averaged) were V(2,2,3,3) and I(1,0,1), respectively. These models also performed significantly better than the other VGGNet-based and InceptionNet-based models in Table 4, with two-sided Wilcoxon rank-sum tests on the 10 runs giving p-values of <0.001. Therefore, this validates our choice of (a,b,c,d) = (2,2,3,3) for the VGGNet-based models and subsystems in our submission. However, it also implies that our choice of (p,q,r) for the InceptionNet-based models and subsystems in our submission was likely to be suboptimal, because choosing I(1,0,1) would have increased the classification accuracy *and* reduced the model complexity simultaneously.

6. CONCLUSION

In conclusion, our submission to DCASE 2020 Task 1B consists of VGGNet- and InceptionNet-based networks either used singularly or combined as an ensemble classifier. Our best performing model used a modified block mixing technique for data augmentation and was pruned to achieve a five-fold reduction in non-zero parameter count. It attained a mean macro-averaged accuracy of 0.9090 (± 0.0020) over 10 runs on the validation set, thus outperforming the baseline. The final performance of the best submitted model on the challenge evaluation set was a mean macro-averaged accuracy of 0.898 with a log-loss of 0.257, which was also better than that of the baseline with a mean macro-averaged accuracy of 0.895 and a log-loss of 0.401. Future work on the topic of low-complexity models could involve developing metrics that encompass the accuracy-complexity dichotomy, possibly in order to find some Pareto-optimal region for accuracy against complexity. Knowledge of such a region would be extremely useful for real-life applications of acoustic scene classification, since users could derive optimal models balancing the desired accuracy against complexity for specific use cases.

7. REFERENCES

[1] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "MixUp: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018, pp. 1–13.

[2] Y. Sakashita and M. Aono, "Acoustic Scene Classification by Ensemble of Spectrograms Based on Adaptive Temporal Divisions," 2018, doi: 10.1109/mra.2018.2802120.

[3] H. Yang, C. Shi, and H. Li, "Acoustic Scene Classification Using CNN Ensembles and Primary Ambient Extraction," 2019.

[4] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2016, pp. 2982–2986, doi: 10.21437/Interspeech.2016-805.

[5] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the Data Augmentation Scheme with Various Classifiers for Acoustic Scene Modeling," 2019, [Online]. Available: http://arxiv.org/abs/1907.06639.

[6] S. Waldekar and G. Saha, "Wavelet-based audio features for acoustic scene classification," 2018.

[7] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, pp. 1–15, 2019, doi: 10.3390/s19071733.

[8] H. Seo, J. Park, and Y. Park, "Acoustic Scene Classification using Various Pre-Processed Features and Convolutional Neural Networks," *Detect. Classif. Acoust. Scenes Events 2019*, pp. 3–6, 2019.

[9] H. Zeinali, L. Burget, and H. Cernosky, "Convolutional Neural Networks and X-vector Embedding for DCASE2018 Acoustic Scene Classification Challenge," 2018.

[10] M. Ebrahimpour *et al.*, "End-to-end Auditory Object Recognition via Inception Nucleus," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2020*, 2020, pp. 146–150.

[11] J. Huang *et al.*, "Acoustic Scene Classification Using Deep Learning-based Ensemble Averaging," in *Detection and Classification of Acoustic Scenes and Events 2019*, 2019, pp. 1–5.

[12] M. D. McDonnell and W. Gao, "Acoustic Scene Classification Using Deep Residual Networks with Late Fusion of Separated High and Low Frequency Paths," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2020*, 2020, pp. 141–145, doi: 10.1109/icassp40776.2020.9053274.

[13] K. Koutini, H. Eghbal-zadeh, G. Widmer, and J. Kepler, "CP-JKU Submissions to DCASE'19: Acoustic Scene Classification and Audio Tagging with Receptive-field-regularized CNNs," in *Detection and Classification of Acoustic Scenes and Events 2019*, 2019, pp. 1–5.

[14] L. Yang, X. Chen, and L. Tao, "Acoustic Scene Classification Using Multi-scale Features," 2018.

[15] J. Jung, H.-S. Heo, H. Shim, and H.-J. Yu, "Knowledge Distillation With Specialist Models in Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2019*, 2019, pp. 5–7.

[16] Y. Lecun, J. Denker, and S. Solla, "Optimal brain damage," 1989.

[17] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2020 3Class, Development dataset," 2020. https://doi.org/10.5281/zenodo.3670185 (accessed Jun. 15, 2020).

[18] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2020 3Class, Evaluation dataset," 2020. https://doi.org/10.5281/zenodo.3685835 (accessed Jun. 15, 2020).

[19] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, 2018, no. November, [Online]. Available: http://arxiv.org/abs/1807.09840.

[20] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2016-May, pp. 6440–6444, 2016, doi: 10.1109/ICASSP.2016.7472917.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations*, 2015, pp. 1–14, [Online]. Available: http://arxiv.org/abs/1409.1556.

[22] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[23] W. Lim, S. Suh, and Y. Jeong, "Weakly Labeled Semi-Supervised Sound Event Detection Using Crnn With Inception Module," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2018, no. November 2018, pp. 74–77.

[24] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "A hybrid approach with multi-channel I-vectors and convolutional neural networks for acoustic scene classification," in *25th European Signal Processing Conference, EUSIPCO 2017*, 2017, vol. 2017-Janua, pp. 2749–2753.

[25] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017, pp. 131–135.

[26] M. H. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," in *6th International Conference on Learning Representations, ICLR 2018*, 2018, pp. 1–14.