ENSEMBLE OF PRUNED MODELS FOR LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFCATION

Technical Report

*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. For this report, 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 trained with log-mel spectrograms of the raw audio data, and were subsequently pruned by setting low-magnitude weights periodically to zero with a polynomial decay schedule for an 80% reduction in individual subsystem size. The resultant ensemble classifier outperformed the baseline model on the validation set over 5 runs and had 119758 non-zero parameters which took up 468KB of memory, thus showing the efficacy of the pruning technique. 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, and 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 account for 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: Data-driven approaches looking to modify or augment the given dataset, representation-driven approaches looking to transform the given raw audio data to a different and possibly more salient form, and model-driven approaches looking to find building blocks and architectures that best replicates the desired output given a particular input. We provide a brief overview of the main techniques we observed as follows.

For data-driven approaches, other than the usage of external data, mixup augmentation [1], [2] 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 [3]. 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 [4].

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 [5], a combination of chroma, spectral contrast, and tonnetz features [6], and separation into harmonic and percussive components [1]. 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 [2], as well as generalized cross-correlation-phase-transform (GCC-PHAT) and interaural time difference (ITD) features [7].

For model-driven approaches, 2-dimensional (2D) convolutional neural network (CNN) classifiers with fully connected layers 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 [8], [9] and AclNet [10]. 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 [11], Su et al. modified an Xception network to allow for prediction with multi-scale features

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

from outputs at different depths [6], and Wan et al. modified ResNet and DenseNet to incorporate receptive-field regularization and frequency-awareness [12]. Other approaches include the application Dempster-Shafer evidence theory to aggregate subsystem outputs into an ensemble classifier [13], as well as the usage of a domain adaptation network to cope with potential device mismatch problems [14]. Lastly, a method that also reduces model complexity is knowledge distillation, where a larger teacher model is used to train a smaller student model to mimic the teacher's outputs [15].

However, the efficacy of the methods mentioned so far have yet to be explored for low-complexity applications. One way to adapt these methods for low-complexity application is to preserve the architecture but omit redundant or low-magnitude parameters in a method known as model pruning. The idea was initially proposed for neural networks by Lecun et al. [21], and could potentially help to ameliorate overfitting problems with complex models by reducing the parameter count as well. Recent approaches to reduce model complexity also combine this with other techniques, albeit in the field of image processing and not audio processing. For instance, Han et al. applied a combination of pruning, quantization and Huffman coding on existing networks for the MNIST and ImageNet datasets [22], and Hooker et al. investigated the effect of pruning on the class-wise accuracy of various image classes on ImageNet models [23].

For our submission, we shall focus on the effect of pruning on model accuracy and complexity for acoustic models, and hence make use of relatively straightforward architectures and data preprocessing methods to observe it.

2. DATA PREPROCESSING

For our submission, we used the TAU Urban Acoustic Scenes 2020 3Class dataset [16], [17], which consists of 10-second long recordings captured with an electret binaural microphone (Soundman OKM II Klassik/Studio A3) and audio recorder (Zoom F8) at a sampling frequency of 48kHz and a depth of 24 bits [18]. The dataset features 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, and the log-mel spectrograms were subsequently generated from the short-time Fourier transform 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. Hence, each spectrogram could be represented by a 48-by-467-by-1 tensor.

2.2. Data Augmentation

We used a simplified version of random block mixing to augment the TAU Urban Acoustic Scenes 2020 3Class dataset. Each augmented track consists of ten 1-second long segments from different recordings in the original dataset that have been concatenated in sequence. The 1-second long segments for each augmented track were chosen at random points of random recordings belonging to the same 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.

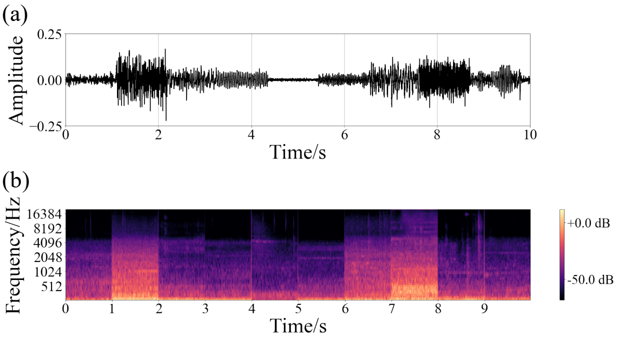

Figure 1: Example of an augmented track with label "transportation" as a (a) time-domain signal and (b) log-mel spectrogram.

3. NETWORK ARCHITECTURE

The networks that we used for our submission to Task 1B are shallower versions of VGGNet [19] and InceptionNet [20]. The choice of shallower networks was made to reduce the number of parameters in the overall model in line with the motivation for Task 1B. The networks use of 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.

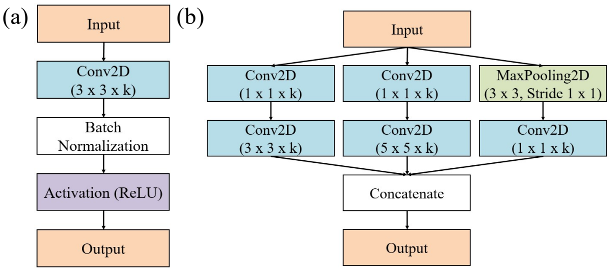

Figure 2: Architecture of (a) VGG(k) module and (b) Inception(k) module.

3.1. VGGNet-based Architecture

The first network that we used was inspired from VGGNet and used VGG(k) modules with increasing numbers of filters in later layers. It contained 80839 parameters in 32-bit floating-point representation, taking up a total of 315.8KB of memory. The architecture is shown in Figure 3.

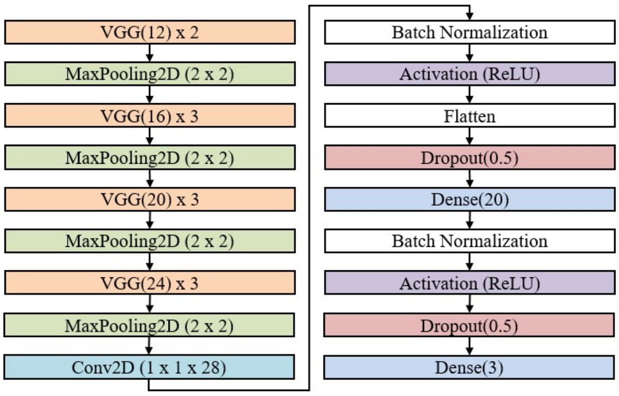

Figure 3: Architecture of VGGNet-based network

3.2. InceptionNet-based Architecture

The second network that we used was inspired from InceptionNet and used Inception(k) modules with increasing numbers of filters in later layers. As advised by the authors in [20] for improved prediction accuracy, we did not perform batch normalization for the Inception(k) modules, and used Inception(k) modules only at the latter layers of the network with regular convolutional layers at the beginning. The overall network contained 167571 parameters in 32-bit floating-point representation, taking up a total of 654.6KB of memory, and its full architecture is shown in Figure 4.

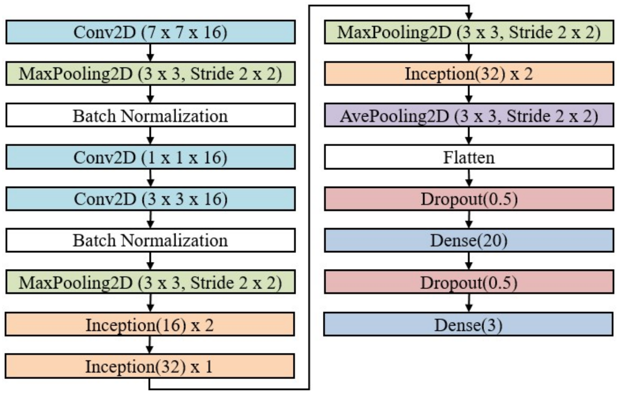

Figure 4: Architecture of InceptionNet-based network.

3.3. Ensemble Classifier

In addition to the two basic networks described in Sections 3.1 and 3.2, 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.

3.4. Submitted Models

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

- Model 1: Single VGGNet-based model trained on non-augmented data.
- Model 2: Single InceptionNet-based model trained on non-augmented data.
- Model 3: Ensemble classifier (five VGGNet-based models and one InceptionNet-based model) trained on non-augmented data.
- Model 4: Ensemble classifier (five VGGNet-based models and one InceptionNet-based model) trained on augmented data.

4. TRAINING METHODOLOGY

Each model (or subsystem) in our submission was trained over 400 epochs with a batch size of 128 samples, with an L2 kernel regularizer (regularization factor 0.001) applied on 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 [24]. The pruning schedule follows a polynomial decay equation as shown in (1). If we denote by s_i the initial sparsity (proportion of model parameters set permanently to zero at the start of the pruning schedule), s_f the final sparsity (proportion of model parameters set permanently to zero at the end of the pruning schedule), n the number of times pruning occurs, t_0 the first epoch when pruning occurs, and Δt number of epochs between each time pruning occurs, then we have

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

for all k in $\{t, t+1, \ldots, t+\Delta t\}$ and t in $\{t_0, t_0+\Delta t, \ldots, t_0+n\Delta t\}$. For our submissions, we used $s_i = 0.1$, $s_f = 0.8$, $n = 20$, $t_0 = 100$, and $\Delta t = 10$. In summary, this gives us a training schedule as outlined in Figure 5.

5. RESULTS AND DISCUSSION

We used the trained 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. Table 1 shows a summary of the performance of the four models described in Section 3.4, and Figure 6 shows the normalized confusion matrices obtained for the best runs of each model.

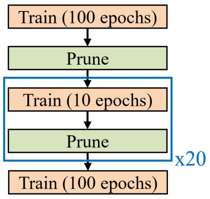

Figure 5: Training and pruning schedule for all networks in the submission. The initial and final sparsity for all models was 0.1 and 0.8 respectively.

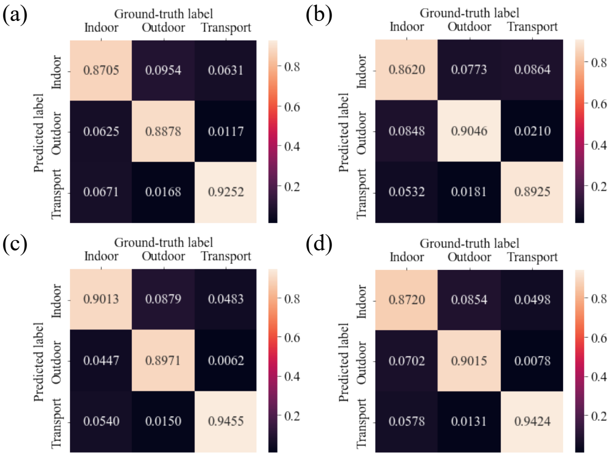

Figure 6: Confusion matrices of the best of 5 runs for (a) Model 1, (b) Model 2, (c) Model 3, and (d) Model 4.

Table 1: Summary of model performance over 5 runs on validation set. Model size was calculated based on number of non-zero parameters.

Model	Mean micro-accuracy	Mean macro-accuracy	# non-zero parameters	Model size/KB
1	0.8841	0.8837	17115	66.9
2	0.8785	0.8783	34181	133.5
3	0.9047	0.9046	119758	467.8
4	0.9050	0.9051	119758	467.8

From Table 1, we can see that all models in our submission exceeded the mean baseline model macro-averaged accuracy of 0.873, which shows that the combination of pruning, shallower models, and modified block mixing could improve classification accuracy for acoustic scene classification tasks as well. All models in our submission were within the size limit of 500KB as well.

With the pruning schedule we described in Section 4, we can also see that both the pruned VGGNet- and InceptionNet-based models (Models 1 and 2) had a five-fold reduction in number of non-zero parameters and model size as compared to the full models. In addition, the ensemble classifiers (Models 3 and 4) performed markedly better than the single models (Models 1 and 2), with a 2-3% increase in both micro-averaged and macro-averaged accuracy over the single models. However, comparing the results from Model 3 and Model 4, it appears that our proposed data augmentation technique only achieves a marginal increase in mean accuracy (both macro and micro), since the only difference in Model 3 and Model 4 is the dataset used to train them. Lastly, we can see from the confusion matrices in Figure 6 that the models tended to perform better in class-wise accuracy for the "transport" class than the "indoor" or "outdoor" classes. This may be due to similar frequency signatures between recordings from the "indoor" and "outdoor" class, since we used log-mel spectrograms (which are inherently frequency-sensitive) as inputs to our model submissions.

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. We used a modified block mixing technique for data augmentation and pruned the networks to achieve a five-fold reduction in non-zero parameter count while outperforming the baseline model in both macro-averaged and micro-averaged classification accuracy. Future work on this could involve comparing the change in performance of the pruned networks against the full networks, as well as to develop metrics that encompass the accuracy-complexity dichotomy, possibly in order to find some Pareto-optimal region for accuracy against complexity.

7. REFERENCES

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

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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] H. Zeinali, L. Burget, and H. Cernosky, "Convolutional Neural Networks and X-vector Embedding for

DCASE2018 Acoustic Scene Classification Challenge," 2018.

[9] 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.

[10] 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.

[11] 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.

[12] M. Wan *et al.*, "CIAIC-ASC System for DCASE 2019 Challenge Task 1," in *Detection and Classification of Acoustic Scenes and Events 2019*, 2019, pp. 3–7.

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

[14] 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.

[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] 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).

[17] 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).

[18] 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.

[19] 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.

[20] 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.

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

[22] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, pp. 1–14, 2016.

[23] S. Hooker, A. Courville, Y. Dauphin, and A. Frome, "Selective Brain Damage: Measuring the Disparate Impact of Model Pruning," 2019. [Online]. Available: http://arxiv.org/abs/1911.05248.

[24] 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.