# DIVERSE SPARSITY SYSTEM USING CONVOLUTION NEURAL NETWORK

Technical Report

*Hui-Hsin Jeng*[1], *Chia-Ping Chen*[1], *Chung-Li Lu*[2], *Bo-Cheng Chan*[2]

[1]National Sun Yat-sen University, Kaohsiung, Taiwan,
m083040026@student.nsysu.edu.tw, cpchen@cse.nsysu.edu.tw
[2]Chunghwa Telecom Laboratories, Taoyuan, Taiwan, {chungli,cbc}@cht.com.tw

## ABSTRACT

In this technical report, we present our works on pruning convolution neural networks and using the quantization method to reduce parameters. DCASE2021 subtask 1A limit classifier size smaller than DCASE2020 subtask 1B with only 128 KB. Therefore we propose three pruning and quantization methods on Convolution Neural Networks. To prune the bigger network ( FCNN ) with single sparsity or diverse sparsity and quantization method. Another proposed method is simply pruning a smaller network ( MobNet ) with single sparsity and quantization method. Our best system performs 1.428 on validation log loss.

***Index Terms—*** acoustic scene classifier, quantization, model pruning, convolution neural network

## 1. INTRODUCTION

Acoustic scene classification is a major task in the Detection and Classification of Acoustic Scenes and Events ( DCASE ) since 2013. Acoustic classification is a major task in the Detection and Classification of Acoustic Scenes and Events ( DCASE ) since 2013. As time goes on, now in 2021, the acoustic scene classification not only needs to classify with multiple devices but also limits the model size. A lighter model can easily deploy on edge devices or cell phones, even integrate with other systems.

With the limitation of none zero parameters only 128 KB, we decide to focus on model pruning and quantization but not built our model. As the open evaluation platform DCASE shows that many acoustic scene classification systems are based on deep neural networks, especially convolution neural networks ( CNN ) [1, 2, 3, 4]. Among the CNN models, the fully convolutional neural network ( FCNN ) and Mobnet ( MobileNetV2 [5] ) are two better models designed in [2]. The origin MobileNet model is designed for edge devices, but not smaller enough for this task. Since the two models are built-in the Keras framework, as long as the official DCASE Baseline system. Choosing the same framework as DCASE made the size calculation easier and other submissions. Experimentally compress the larger model FCNN first, then compress the smaller model Mobnet.

Different model compression methods such as weight clustering, post-training quantization, model pruning[6], or binary convolutions can compress the model in their ways. Post-training quantization not only is one of the easiest ways to reduce model parameters but also remains flexible into several different precision. 128 KB is the limitation of the acoustic scene classifier to submit. Namely Mobnet with post-training quantization would be bigger than the limitation, much less FCNN with post-training quantization. Therefore, model pruning would be capable of compressing most model parameters. While pruning the model, training the rest of the parameters made the minimal reduction of log-loss or accuracy.

The rest of this report is organized as follow. Section 2 presents different models and details. Section 3 explains the pruning method. Post-training quantization details are described in Section 4. Experimental setup and results are described in Section 5. We draw conclusion from the results in Section 6.

## 2. ACOUSTIC SCENE CLASSIFIER

Two acoustic scene classifiers we submitted are both CNN structure, the structure and compression methods are shown in the follows.

- Fully convolution neural network ( FCNN ) consists 9 convolution layers and 2 fully connected layers. The origin FCNN model size is 509,598.6 KB, with the default 32 bit floating point precision. The submitted FCNN model was pruned with details shows in Section 3. The precision of convolution layers is set to float 16 ( Section 4 ).
- Mobnet ( MobileNetV2 ) is based on the Inverted residual block. However smaller then MobileNetV2 for satisfaction of the task rules. Since Mobnet is a smaller model, the sparsity of model pruning is set smaller then FCNN. The precision is set to int 8.

## 3. MODEL PRUNING

We use the TensorFlow framework for model pruning. Each chosen layer is separately managed. Adding a binary mask determines the weights participation in forward execution. Then mask the layer's absolute values until the desired sparsity $s$. The masked weights do not update in the back-propagation. Equation 1 is the pruning algorithm, the system sparsity increases from the initial sparsity value $s_i$ to final sparsity $s_f$ by $n$ pruning step. We need to set the start training step $t_0$ and pruning frequency $\Delta t$.

$$s_t = s_f + (s_i - s_f)(1 - \frac{t - t_0}{n\Delta t})^3 for\ t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\} \quad (1)$$

With the above algorithm, we gradually increase the final sparsity. Training different sparsity FCNN models to see the sparsity increase and accuracy cost. Figure 1 shows that when we prune the FCNN system for more sparsity, costs more accuracy loss. The

Table 1: Results of evaluating model with different sparsity level and precisions. The submission number represents the submitted system numbers.

| Model | Submission | Prune Convolutions | Prune Others | Quantization | Non Zero Parameters ( KB ) | Log Loss | Accuracy |
|---|---|---|---|---|---|---|---|
| 2021 task1a Baseline | - | - | | Float 16 | 90.3 | 1.551 | 0.41 |
| FCNN | - | - | | - | 509,598.6 | 1.182 | 0.70 |
| | 1 | 0.99479 | | Float 16 | 126.3 | 1.464 | 0.54 |
| | 2 | 0.9999 | 0.99485 | Float 16 | 124.9 | 1.593 | 0.51 |
| Mobnet | - | - | | - | 67,136.5 | 1.417 | 0.58 |
| | 3 | 0.48 | | Int 8 | 127.9 | 1.428 | 0.58 |
| | - | 0.97 | | - | 134.9 | 2.017 | 0.27 |

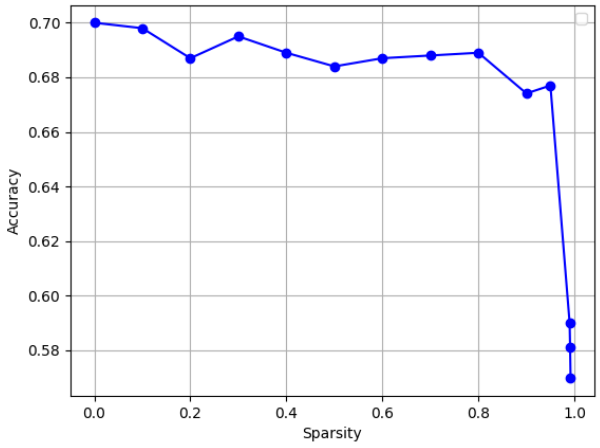


Figure 1: Training different sparsity of FCNN. This figure shows that when model sparsity increases, cost the system of accuracy loses.

best option to prune a model should be the point before the accuracy drops. In our experiments, we found that after pruned to the final sparsity, keep training for some epoch helps the pruned model performance.

### 3.1. Diverse Sparsity System

The diverse sparsity system uses the same algorithm in Equation 1 but sets different sparsity for different layers. Most of the parameters in FCNN belong to convolution layers. For the above reason, we increase the sparsity of convolution layers in FCNN. However, set the remaining layers with smaller sparsity. Namely, prune convolution layers more for keeping other layers.

## 4. QUANTIZATION

TensorFlow platform provides a post-training quantization technique, TFLite, to reduce the model size by reducing precision. For the supported layers, reduce the precision to Float 16 for about 1/2 size reduction. For 8-bit quantization, helps reduced to 1/4 size of the original model.

## 5. EXPERIMENTS

### 5.1. Features

TAU Urban Acoustic Scenes 2020 Mobile, development dataset provides fixed-length 10-second audios. Using log-mel filter bank, we use the Librosa tool [7] generate with 2048 STFT points, 2048 samples, and 1024 frameshift. The frequency is 128. Then compute Log-mel deltas and delta-deltas without padding, with the result of input tensor 423x128x3. Last but not least, normalize each feature value.

### 5.2. Training Setup

We refer to the most configuration as the author of FCNN. Train both FCNN and MobNet for 500 epoch. During pruning, we train only 180 epochs, but not 500. Start the pruning from the beginning step and update the binary mask every 200 steps. Finish the pruning at 170 epoch but keep training until 180 epoch. Follow the official recommended way of splitting the dataset. There are 13962 training clips and 2968 testing clips from different devices. During training, we use a mixup augmentation method at the mini-batch level.

### 5.3. Results And Submission Summary

Original FCNN and MobNet we trained and the compressed systems are shown in Table 1, as well as 2021 Baseline system for comparison. With the compression methods, the accuracy and loss preform reduced.

- Submission 1 : DCASE2021 Sparse FCNN System
  We prune the FCNN model till the final sparsity is 0.99479, as explained in Section 3. Then convert the precision into 16 bits. The final non-zero parameters remain only 126.3 KB, reducing 0.9997 bytes.
- Submission 2 : DCASE2021 Diverse Sparsity FCNN System
  Section 3.1 shows the detail of diverse sparsity system. We reduce the convolution layers into the minimum parameters, in the meantime prune other layers as well. Finally, convert precision with post-training quantization into float 16.
- Submission 3 : DCASE2021 Sparse MobNet System
  Since MobNet is a smaller system, we set the final sparsity bigger. Thus, set the quantization method with smaller precision, int 8. Reducing 0.9979 bytes.

## 6. CONCLUSION

This technical report shows the detail that we tackle Task 1A of 2021 DCASE. Training a large classifier is not an abstruse problem. However, limiting the model size raise all other problems. Our compression method in the report shows that we reduce 99.97% of the complexity. The best of our systems has an accuracy of 0.58 and log loss of 1.428, which is a little performance loss.

## 7. REFERENCES

[1] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with RF-regularized CNNs," DCASE2020 Challenge, Tech. Rep., June 2020.

[2] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," DCASE2020 Challenge, Tech. Rep., June 2020.

[3] M. McDonnell, "Low-complexity acoustic scene classification using one-bit-per-weight deep convolutional neural networks," DCASE2020 Challenge, Tech. Rep., June 2020.

[4] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," DCASE2020 Challenge, Tech. Rep., June 2020.

[5] S. Mark, H. Andrew, Z. Menglong, Z. Andrey, and C. Liang-Chieh, "Mobilenetv2: Inverted residuals and linear bottlenecks," *arXiv preprint arXiv:1801.04381*, Macrch 2019.

[6] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *arXiv preprint arXiv:1710.01878*, October 2017.

[7] librosa, "librosa," https://github.com/librosa/librosa, 2020.