CAU-ET SUBMISSION TO DCASE 2021: TRIDENT-EFFICIENTNET FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Soyoung Lim[1], *Yerin Lee*[1], *Il-Youp Kwak*[1]

[1] Chung-Ang University, Department of Applied Statistics,
Seoul, South Korea, {isy921,dldpfls14,ikwak2}@cau.ac.kr

ABSTRACT

Acoustic scene classification (ASC) categorizes an audio file based on the environment in which it has been recorded. This has long been studied in the detection and classification of acoustic scenes and events (DCASE). We present the solution to Task 1 A (Low-Complexity Acoustic Scene Classification with Multiple Devices) of the DCASE 2021 challenge submitted by the Chung-Ang University team. We proposed Trident-EfficientNet with 3 scaling factors: width, depth, resolution. Additionally, we used lightweight deep learning techniques such as pruning and quantization.

Index Terms— acoustic scene classification, efficientnet, pruning, quantization

1. INTRODUCTION

The goal of Task 1 in DCASE 2021 is to classify a test recording into one of the provided predefined classes that characterize the acoustic scenes in which it was recorded [1]. We submitted the results for subtask A of Task 1. The subtask A addressed two challenges that ASC faces in real-world applications. Both are concerned with the basic problem of acoustic scene classification. One is that the audio recorded using different recording devices should be classified in general, and the other is that the model used should have low-complexity. Subtask A's audio data are recorded and simulated with a variety of devices. The development dataset comprises 40 hours of data from device A, and smaller amounts from the other devices. Audio is provided in single-channel (mono) 44.1kHz 24-bit format.

2. ARCHITECTURE

2.1. System Overview

Figure 1 describe our system overview. We extract log-mel spectrogram, delta, and delta-delta features from the raw audio. Combined feature of log-mel spectrogram, delta, and delta-delta fed into our model. In model training, we trained our proposed model with mixup and crop augmentation. We further applied pruning and quantization for the composition of even lighter models.

2.2. Audio Preprocessing

In the past DCASE challenges, most of the top team approached to forming image like spectrograms as inputs for Convolutional Neural Networks (CNN). For feature extraction, our approach was inspired by McDonnell's past work on DCASE 2019 competition [2], that utilize log-mel energies, deltas, and delta-deltas from the log-mel energies. The deltas and delta-deltas imply the first and second temporal derivatives of the spectrum. The audios in the subtask A are mono and have a common sampling rate of 44.1kHz. To generate each spectrogram, we used 2048 FFT points, a hop-length of 1024 samples, 300 frequency bins and HTK formula. The stacked feature of log-mel spectrogram, deltas, and delta-deltas fed into our deep learning models.

2.3. Data Augmentation

Mixup is an effective data augmentation method [3]. We used a general augmentation approach: we mixed different samples of the training set according to their weights. The method is as follows:

$$X = \lambda X_i + (1-\lambda)X_j, \tag{1}$$

$$y = \lambda y_i + (1-\lambda)y_j, \tag{2}$$

where $\lambda \in [0,1]$ and is acquired by the sampling of the beta distribution with parameter α, $\beta(\alpha,\alpha)$, $\alpha \in (0,\infty)$. X_i and X_j are different data samples ; y_i and y_j are their corresponding labels. In our experiment, we used the mixup to augment the log-mel-spectrograms. We set α at 0.4 and used crop augmentation as 300 on the temporal axis before the mixup augmentation [2].

2.4. Trident EfficientNet

CNN have the transition invariant property meaning that we can detect an objective no matter where it located. A cat on top left, and a cat on bottom right are same as a cat category of images. However in spectrogram, invariant property in frequency domain is not quite necessary. Low frequency regions and high frequency regions would have its own meaning in audio domain. Thus, we took the trident architecture proposed by Suh et al. (2020) [4], dividing frequency ranges in three parts in their modeling architecture. The Figure 2 (a) describes the whole modeling architecture. The input feature passed to the 3×3 convolution layer with 32 filters. The frequency domain of the output is equally divided into three parts. The original (150, 212,3) dimension is divided into three (50,212,3) dimensions. Each (50,212,3) dimension path, we connected three Efficient Blocks. The Efficient Block is described in Figure 2 (c), and the MB-SE block is described in Figure 2 (b). The Efficient

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (2020R1C1C1A01013020)

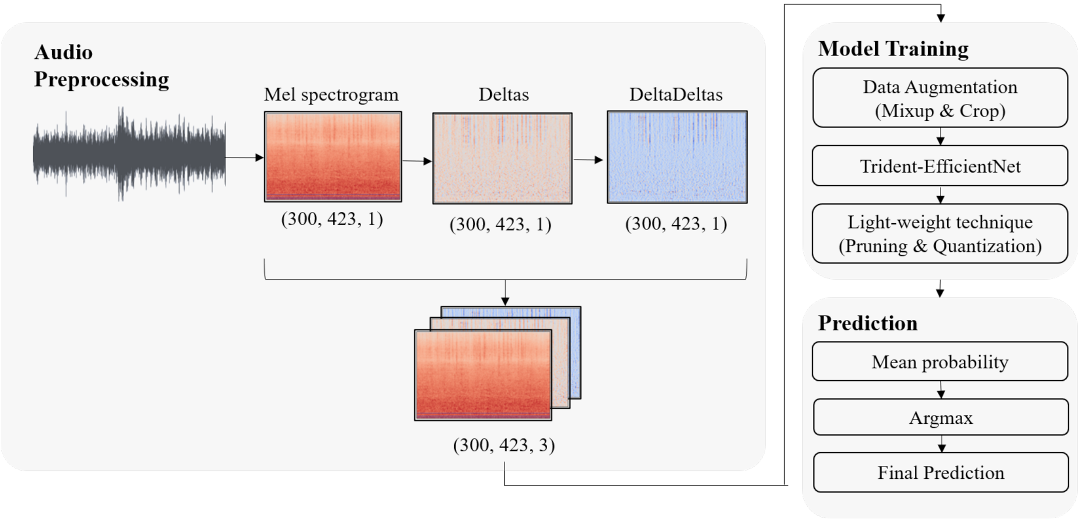
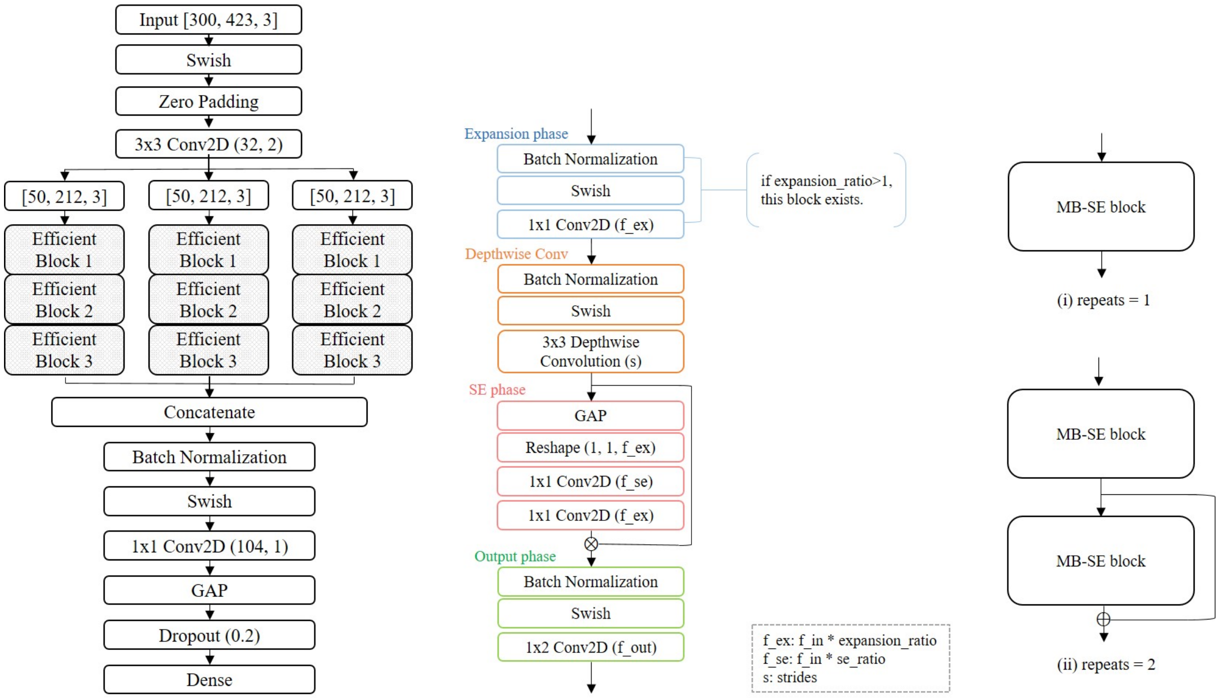


Figure 1: System Overview



(a) Trident-EfficientNet (b) MB-SE block (c) Efficient block

Figure 2: Model architecture. (a) describe whole model architecture. (b) describe MB-SE Block used in the model. (c) describe Efficient Block used in the model (a). If the repeats parameter is 1, (i) is used as Efficient block. If the repeats parameter is 2, (ii) is used as Efficient block. The repeats parameter is described in Table 1, 2, and 3. f_in represent filters in and f_out represent filters out.

block is inspired by Efficientnet [5] modifying the architecture of Efficientnet-B0 model. We revised Efficient Block in the order of batch normalization layer, activation layer and convolution layer. Also, the kernel size of the convolution kernel used in the output phase of the MB-SE block was modified from (1,1) to (1,2). The MB-SE block consists of expansion phase, depthwise convolution, squeeze-and-excitation (SE), and output phase. The SE layer is added in the inverted bottleneck layers as Xiong et al (2019) [6] suggested in their ANTNets. In the expansion phase (in the inverted bottleneck), The number of output channels increased by the input channel (filter in) multiplied by the expand ratio. Next, the depthwise separable convolution is applied with a SE layer. In the output

Table 1: Efficient Block parameters for TEFF 1 model

Block	repeats	filters in	filters out	expand ratio	strides	se ratio
1	1	32	16	1	1	0.25
2	2	16	24	3	2	0.25
3	2	24	32	2	1	0.25

Table 2: Efficient Block parameters for TEFF 2 model

Block	repeats	filters in	filters out	expand ratio	strides	se ratio
1	1	32	16	1	1	0.25
2	2	16	24	3	2	0.25
3	2	24	40	3	1	0.25

phase, the convolution with the 1×2 kernel is applied.

We experimented with four different systems by modifying Efficient Block parameters. The Table 1 describe Efficient Block parameters for TEFF 1 model. The Table 2 and the Table 3 describe Efficient Block parameters for TEFF 2 and TEFF 3 model. The Figure 2 (c) (i) describe the Efficient Block when the repeats parameter is 1, and The Figure 2 (c) (ii) describe the Efficient Block when the repeats parameter is 2,

3. EXPERIMENTS

3.1. Datasets

The development dataset of TAU Urban Acoustic Scene 2020 Mobile [7] were collected by the Tampere University of Technology (TAU) between May and November 2018. The dataset contains recordings obtained from 10 European cities using 9 different devices: 3 real devices (A, B, and C) and 6 simulated devices (S1–S6). A recording from device A is processed through convolution with the selected impulse response, then processed with a selected set of parameters for dynamic range compression (device-specific). The development dataset consists of total 23,000 samples, the dataset is provided with a training and test datasets in which 70% of the data for each device is included for training and 30% for testing. Some simulated devices (S4–S6) appear only in the test subset. As a result, we have 13,962 training samples and 2,970 testing samples.

We divided the original training data by 7 to 3 for the new training and validation set. Our model was trained using the new training set and validation accuracy is measured on the validation set. The final scores are calculated using all the original training dataset.

3.2. Experiment Setting

Each model trained using 70 epochs with Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9. The stochastic gradient descent with warm restarts is used [8]. Learning rate was initially set to 10^{-2} and decreased to 10^{-5}. We used cosine decay warm restart, and it is initialized at epoch number 10 and 30.

Table 3: Efficient Block parameters for TEFF 3 model

Block	repeats	filters in	filters out	expand ratio	strides	se ratio
1	1	32	16	1	1	0.25
2	2	16	24	2	2	0.25
3	1	24	40	2	1	0.25

Table 4: Model size calculation. The Parameters column represent the number of parameters and the NZ parameters column represent the number of non-zero (NZ) parameters. The model size described inside of the parenthesis.

System Name		Parameters	NZ parameters
TEFF1-C45-Q	32bit	7,628 (29.8KB)	7,628 (29.8KB)
	16bit	82,282 (160.7KB)	48,871 (95.45KB)
	Total	89,910 (190.5KB)	56,499 (125.2KB)
TEFF1-P45-Q	32bit	7,628 (29.8KB)	7,628 (29.8KB)
	16bit	82,282 (160.7KB)	48,871 (95.45KB)
	Total	89,910 (190.5KB)	56,499 (125.2KB)
TEFF2-C70-Q	32bit	9,676 (37.8KB)	9,676 (37.8KB)
	16bit	125,072 (244.3KB)	44,828 (87.55KB)
	Total	134,748 (282.1KB)	54,504 (125.4KB)
TEFF3-Q	32bit	4,780 (18.67KB)	4,780 (18.67KB)
	16bit	51,266 (100.1KB)	51,266 (100.1KB)
	Total	56,046 (118.8KB)	56,046 (118.8KB)

3.3. Light-weight Techniques

We applied two types of pruning schedulers. One is constant sparsity that only uses target sparsity, the other is the polynomial decay which increases the sparsity from 0 to target sparsity amount. We applied pruning for the first 30 epochs and we fine tuned the initial learning rate with the second phase of 30 epochs of training. [9]

We proposed four different systems. Table 5 shows the name of our submitted systems and configurations for pruning and quantization. ‘TEFF1-C45-Q’ and ‘TEFF1-C45-Q’ used the same TEFF 1 model. The ‘TEFF1-C45-Q’ model have constant sparsity level of 45%, and the ‘TEFF1-P45-Q’ model have polynomial decay sparsity that the sparsity level increases from 0 to 45%. For the ‘TEFF2-C70-Q’ model, we experimented more complex model with increased level of sparsity. Lastly, ’TEFF3-Q’ model, we used light model without pruning. The original four model parameters were expressed in 32 bit format. We used 16 bit quantization except for batch normalization layers. The detailed model size calculation is shown in the Table 4.

3.4. Model Configurations

The naming convention for our proposed model described below:

- TEFF: ‘TEFF’ indicate that we used Trident-EfficientNet model. We used different hyper parameters for our proposed models.
- C: ‘C’ indicate that we used constant sparsity for pruning scheduler. The number after ‘C’ letter indicate the sparsity level.
- P: ‘P’ indicate that we used polynomial decay for pruning scheduler. The number after ‘P’ letter indicate the target sparsity level.
- Q: ‘Q’ indicate that we used 16bit Quantization.

4. RESULTS

In this section, we report the performance of our proposed models on the validation set separated from the original training dataset. The separated validation set is not used in the model training. In

Table 5: Light-weight techniques for Submitted Systems

	System Name	Model	Pruning	Quantization
1	TEFF1-C45-Q	TEFF 1	constant, 45 %	float16
2	TEFF1-P45-Q	TEFF 1	poly decay, 45 %	float16
3	TEFF2-C70-Q	TEFF 2	constant, 70 %	float16
4	TEFF3-Q	TEFF 3	-	float16

Table 6: Performances on validation dataset

Model	System Name	Accuracy	Size
1	TEFF1-C45-Q	65.5 %	125.1KB
2	TEFF1-P45-Q	65.7 %	125.1KB
3	TEFF2-C70-Q	65.2 %	125.4KB
4	TEFF3-Q	63.1 %	118.8KB

all results,it exceeds the accuracy of the Baseline system, and all models have a model size of less than 128 KB.

4.1. Our submission

We submitted 1st, 2nd and 3rd model in Table 6 for subtask A.The Table 7 describe detailed information for our submissions. All submissions are trained using all development dataset.

5. CONCLUSION

We propose the Trident-EfficientNet architecture that maintains the frequency related information with its trident shaped architecture. Also, Efficient block design is applied for light models. Further, we applied pruning and quantization. We experimented with three different Trident-EfficientNet models with different pruning parameters. Our proposed models have accuracy ranges from 63.1% to 65.7% on the separated validation dataset.

6. REFERENCES

[1] http://dcase.community/challenge2021/.

[2] W. Gao and M. McDonnell, “Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths,” DCASE2019 Challenge, Tech. Rep., June 2019.

[3] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.

[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] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[6] Y. Xiong, H. J. Kim, and V. Hedau, “Antnets: Mobile convolutional neural networks for resource efficient image classification,” *arXiv preprint arXiv:1904.03775*, 2019.

[7] T. Heittola, A. Mesaros, and T. Virtanen, “TAU Urban Acoustic Scenes 2020 Mobile, Development dataset,” Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3670167

[8] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[9] A. Renda, J. Frankle, and M. Carbin, “Comparing rewinding and fine-tuning in neural network pruning,” *arXiv preprint arXiv:2003.02389*, 2020.

Table 7: Description for Subtask A submissions. Model is described in Table 6.

Submission ID	Model	Training
Lim_CAU_task1a_1	1	all
Lim_CAU_task1a_2	2	all
Lim_CAU_task1a_3	3	all
Lim_CAU_task1a_4	4	all