LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION USING MOBILE INVERTED BOTTLENECK BLOCKS

Technical Report

Sergey Verbitskiy

Deepsound
Novosibirsk, Russia
s.verbitskiy@yahoo.com

Viacheslav Vyshegorodtsev

Deepsound
Novosibirsk, Russia
vyshegorodtsevslava@gmail.com

ABSTRACT

This technical report describes our approaches for Task 1A (Low-Complexity Acoustic Scene Classification with Multiple Devices) of the DCASE 2021 Challenge. We propose a new architecture with mobile inverted bottleneck blocks (Fused-MBConv and MBConv) for acoustic scene classification tasks. This architecture is based on EfficientNetV2. Our models have a very small number of parameters. We also use several data augmentation techniques during the training of models. Our best model has 62,346 non-zero parameters and achieves a classification macro-average accuracy of 70.5% and an average multiclass cross-entropy (log loss) of 0.848 on the development dataset. The resulting model size is 121.8 KB (the model parameters are quantized to float16 after the training).

Index Terms— acoustic scene classification, mobile inverted bottleneck blocks, data augmentation techniques.

1. INTRODUCTION

In this work, we describe our approaches and systems for acoustic scene classification (ASC) tasks. ASC is a subtask of audio pattern recognition task and is an important topic in machine learning and signal processing areas because audio signals can contain a lot of rich information.

The Detection and Classification of Acoustic Scenes and Events 2021 Task 1A [1] focuses on the robustness to various devices and low-complexity of ASC systems. This is important because, firstly, environmental sounds are recorded on various devices, and secondly, in real life, it is necessary to efficiently process a large number of audio streams from many devices. For DCASE 2021 Task 1A the size of the ASC system must be limited to 128 KB for the non-zero parameters. This equals 32,768 parameters for float32 format and equals 65,536 parameters for float16 format.

Systems with convolutional neural networks (CNNs) outperformed systems with another approaches for audio pattern recognition tasks and are used most often [2], [3], [4], [5]. These systems use different 2D input features such as log-mel spectrogram. Our ASC systems also is based on CNNs.

In the work [6] (EfficientNet) was proposed convolutional neural networks scaling strategy using network width, depth, and resolution of input features. Using this scaling strategy, in [6] it was demonstrated that EfficientNet models can be simply scaled, are very efficient, and have high classification accuracy for image classification tasks. In EfficientNetV2 [7] was used an idea for replacement of some MBConv [8] (main convolutional blocks in EfficientNet) blocks with another mobile inverted bottleneck blocks (Fused-MBConv) [9] and were applied the most modern approaches for image classification tasks. As a result, it was possible to achieve better accuracy, reduce training and inference time of CNNs.

We propose a new architecture, which is based on EfficientNetV2 [7], with mobile inverted bottleneck blocks (Fused-MBConv and MBConv) for very efficient ASC systems. We also apply several augmentation techniques during training of models.

Our best system has 62,346 non-zero parameters and achieves a classification macro-average accuracy of 70.5% and an average multiclass cross-entropy (log loss) of 0.848 on the development dataset. The resulting model size is 121.8 KB (the model parameters are quantified to float16 after the training).

2. ACOUSTIC SCENE CLASSIFICATION SYSTEM

2.1. Feature extraction

In this paper, we use log-mel spectrograms (log-mel energies) as input time-frequency features to our models. For extracting log-mel spectrograms we adopt a hop size of 690 samples and a window size of 2760 samples (75% overlap) with the Hann window function for the calculating of STFT (Short-Time Fourier Transform). A sampling rate $sr = 44100$ Hz and a signals duration $t = 10$ seconds are fixed for TAU Urban Acoustic Scene 2020 Mobile dataset [10] (we do not use resampling methods).

For our ASC systems, we experiment with a different number of mel bins $M \in \{32, 64, 128, 160\}$. We also determine the lower cut-off frequency $f_{min} = 10$ Hz to remove low-frequency noise and the upper cut-off frequency $f_{max} = 16000$ Hz to remove the aliasing effects.

Thus, the size of input features is equal to $640 \times M$.

2.2. Data augmentation techniques

Several data augmentation techniques are applied to prevent models from overfitting during training:

- **temporal cropping**: models are used 8-second sections of audio signals during training. Sections are cut from random places. During evaluating full audio signals are used as input;
- **SpecAgment** [11];
- **modified mixup** [5]: this mixup takes into account the sound pressure level of two audio signals that are mixed. We adopt $\alpha = 0.6$ for all experiments.

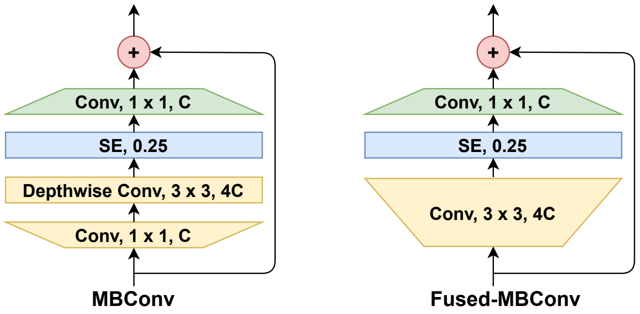

Figure 1: Structure of MBConv4 [8] and Fused-MBConv4 [9] blocks. C is the number of output channels. SE is the squeeze-and-excitation block [12].

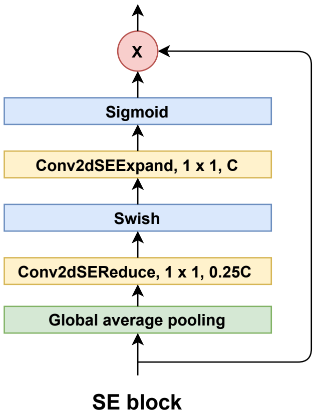

Figure 2: Structure of SE block [12].

2.3. Architecture of ASC models

Our architecture of convolutional neural networks is based on EfficientNetV2 [7]. EfficientNet [6] is a modern neural architecture for efficiency models for image classification tasks. These models have a good trade-off between computational complexity and the performance with optimal choice of the dependence of the width of the convolutional layers, the depth of the neural network, and the size of the input features.

In EfficientNetV2 were added Fused-MBConv blocks [9]. In EfficientNet only MBConv blocks [8] are used, which includes depthwise convolutions. In EfficientNetV2 it was shown that replacement of some MBConv4 blocks with Fused-MBConv4 blocks allows for increasing the performance and efficiency of models. The comparison of MBConv4 with Fused-MBConv4 is shown in Fig. 1.

Structure of the squeeze-and-excitation block (SE) [12] is described in Fig. 2.

Our architecture of ASC models is described in Table 1.

Batch normalization [13] (over frequency axis) is used at the start as replacement to data standardization. We apply only two Fused-MBConv4 blocks at the start because these blocks have a large number of parameters for high values of C (the number of output channels). In the table, stride sizes are described for only the first layers in blocks (stride size is equal to 1×1 for other layers). For global pooling, we use a combination of max pooling and average pooling as in [3] to combine their advantages. After global pooling, we use a fully connected layer (FC) and a softmax nonlinearity at the end to obtain model predictions. As an activation function, we use the swish function [14].

Table 1: Proposed architecture of our ASC system.

Blocks/Layers	Stride	Kernel	Channels
BatchNorm	–	–	1
Conv2d	1×1	5×3	6
2 × Fused-MBConv4	2×2	3×3	12
3 × MBConv4	2×1	3×3	18
3 × MBConv4	2×2	3×3	24
2 × MBConv4	2×2	3×3	30
Conv2d	1×1	1×1	100
Global pooling	–	–	100
Flatten	–	–	100
Fully connected (FC)	–	–	-
Softmax	–	–	-

It is worth noting that we adopt the optimal width of convolutional layers, the number of convolutional layers (depth) per block, stride sizes, and other hyper-parameters by many experiments. We also select these hyper-parameters in such a way as to satisfy the limit of 65,536 parameters. But the results of detailed enumeration of hyper-parameters are omitted. The best values of these hyper-parameters are described in Table 1.

3. EXPERIMENTS AND RESULTS

3.1. Training setup

Parameters of models are optimized by minimizing a categorical cross-entropy loss with the AdamW optimizer [15] with standard parameters and with a batch size of 64. We evaluate models using an exponential moving average of models parameters with a decay rate of 0.999. We also use a one-cycle learning rate policy [16] with a max learning rate of 0.01. All models are trained for about 400 epochs (we use early stopping with max epochs of 500).

3.2. Task 1A: TAU Urban Acoustic Scene 2020 Mobile

TAU Urban Acoustic Scene 2020 Mobile dataset [10] is proposed for systems training and evaluation in DCASE 2021 Task 1A. The development dataset consists of 23,040 audio clips and 10 classes of acoustic scenes. The dataset is balanced by classes.

It is worth noting that this dataset consists of audio clips which were collected from nine devices (three real devices and six simulated devices). For training and evaluation of our systems, we use basic metadata of training/test split which was provided by organizers of the DCASE 2021 Task 1A challenge [1]. The test set is balanced by devices (329-330 clips per device) and contains 2,970 audio clips. The test set also contains three devices (S4 S5, S6) that do not contain in the training set.

For comparison of models, we use two official metrics – macro-average accuracy (average of the class-wise accuracies) and macro-average multiclass cross-entropy (log loss, average of the class-wise log loss). All results of models for the test set of the development dataset are represented for float16 format of parameters (quantization of models is done using PyTorch 1.7 [17] after training and the weights of models are converted to float16 format).

3.3. Comparison of models with different number of mel bins

We compare the performance of models with a different number of mel bins $M \in \{32, 64, 128, 160\}$. Results are shown in Table 2.

Table 2: Comparison of models with a different number of mel bins.

Mel bins	Accuracy	Log loss
32 bins	64.4%	1.042
64 bins	68.8%	0.932
128 bins	**70.9%**	0.859
160 bins	70.5%	**0.848**

Our best model (by a value of log loss) with 160 mel bins achieves a macro-average accuracy of 70.5% and log loss of 0.848. It is worth noting that the model with 64 mel bins has almost the same performance as the model with 160 mel bins, but the model with 64 mel bins is $\approx$ 2.5x faster.

3.4. Class-wise and device-wise performance of the best model

Class-wise and device-wise performance on the test set of the development dataset of our best model (with 160 mel bins) is described in Table 3 and Table 4, respectively.

Table 3: Class-wise performance of the best model.

Scene label	Accuracy	Log loss
Airport	61.5%	1.018
Bus	90.6%	0.340
Metro	71.7%	0.713
Metro station	73.4%	0.764
Park	86.2%	0.518
Public square	51.5%	1.460
Shopping mall	71.4%	0.845
Street, pedestrian	32.3%	1.812
Street, traffic	87.5%	0.421
Tram	78.7%	0.592
Average	**70.5%**	**0.848**

Table 4: Device-wise performance of the best model.

Device	Accuracy	Log loss
A	78.5%	0.627
B	72.3%	0.828
C	77.5%	0.699
S1	68.5%	0.886
S2	67.0%	0.914
S3	69.7%	0.831
S4	68.5%	0.949
S5	67.6%	0.907
S6	64.8%	0.995
Average	**70.5%**	**0.848**

It is worth noting that there is enough difference for class-wise performance for various acoustic scene classes. For example, our model has 90.6% macro-average accuracy for the ”bus” class and 32.3% macro-average accuracy for the ”street, pedestrian” class.

There is a slight difference in the device-wise performance for various devices. For unseen devices S4, S5, and S6 our model has almost the same accuracy as for seen devices S1, S2, S3. Thus, our model is quite robust to various devices, which is one of the main goals of DCASE 2021 Task 1A.

3.5. Model size

In this subsection, we provide full information about the model size. Our model has a large number of layers; therefore, we describe the model size using two tables.

A full description of the layers of Fused-MBConv4 and MBConv4 blocks and the formulas for calculating the number of parameters for each layer are presented in Table 5.

Table 5: The number of parameters of MBConv4(I, O) and Fused-MBConv4(I, O) blocks.

Layer	MBConv4	Fused-MBConv4
Conv2dExpand	$4{\cdot}I^2$	$36{\cdot}I^2$
BatchNorm + swish	$8{\cdot}I$	$8{\cdot}I$
Conv2dDepthwise	$36{\cdot}I$	–
BatchNorm + swish	$8{\cdot}I$	–
Global pooling	0	0
Conv2dSEReduce	$(1 + 4{\cdot}I) \cdot [0.25 \cdot I]$	$(1 + 4{\cdot}I) \cdot [0.25 \cdot I]$
Swish	0	0
Conv2dSEExpand	$4{\cdot}I \cdot (1 + [0.25 \cdot I])$	$4{\cdot}I(1 + [0.25 \cdot I])$
Conv2dReduce	$4{\cdot}I \cdot O$	$4{\cdot}I \cdot O$
BatchNorm2d	$2{\cdot}O$	$2{\cdot}O$
Conv2dShortcut	$H(I, O)$	$H(I, O)$

In the table I is the number of input channels, O is the number of output channels, $[x]$ is the integer part of x and the function $H(x, y)$ is calculated by the formula:

$$H(x, y) = \begin{cases} x \cdot y & x \neq y \\ 0 & x = y \end{cases} \tag{1}$$

Hence, the total number of parameters of MBConv4 block is calculated by the formula:

$$\begin{aligned} N(I, O) = 4 \cdot I \cdot (I + O + 2 \cdot [0.25 \cdot I] + 14) + \\ + 2 \cdot I + [0.25 \cdot I] + H(I, O) \end{aligned} \tag{2}$$

And the number of parameters of Fused-MBConv4 block:

$$\begin{aligned} N(I, O) = 4 \cdot I \cdot (9 \cdot I + O + 2 \cdot [0.25 \cdot I] + 3) + \\ + 2 \cdot I + [0.25 \cdot I] + H(I, O) \end{aligned} \tag{3}$$

In Table 6 we provide the description of the model structure and the number of parameters corresponding to its size in KB. In order not to overload the table there is no column ”non-zero parameters” because the number of non-zero parameters is equal to the number of parameters (we do not use pruning methods). We also do not describe column ”data type” because the data type of each parameter is float16 during evaluation (the model parameters are quantized to float16 after the training).

The total size of our best model (with 160 mel bins) is 121.8 KB, and the model is consistent with the competition requirements.

Table 6: Model size calculation (for 160 mel bins). Size of parameters is described for float16 format.

Block/Layer	Parameters	Size
BatchNorm	320	640 B
Conv2d	96	192 B
BatchNorm + swish	12	24 B
Fused-MBConv4(6, 12)	1,801	3,602 B
Fused-MBConv4(12, 12)	6,219	1,2438 B
MBConv4(12, 18)	2,655	5,310 B
MBConv4(18, 18)	4,216	8,432 B
MBConv4(18, 18)	4,216	8,432 B
MBConv4(18, 24)	5,092	10,184 B
MBConv4(24, 24)	7,158	14,316 B
MBConv4(24, 24)	7,158	14,316 B
MBConv4(24, 30)	8,466	16,932 B
MBConv4(30, 30)	10,627	21,254 B
Conv2d	3,100	6,200 B
BatchNorm + swish	200	400 B
Global pooling	0	0 B
Flatten	0	0 B
Fully connecte (FC)	1010	2,020 B
Softmax	0	0 B
Total	**62,346**	**121.8 KB**

3.6. Results

Table 7 shows results of our systems with different submission ID and the comparison with the baseline system [18] on the official development dataset [1] [10].

- **submission 1**: the model with 32 mel bins;
- **submission 2**: the model with 64 mel bins;
- **submission 3**: the model with 128 mel bins;
- **submission 4**: the model with 160 mel bins;

Table 7: Results for Task 1A on official development data.

Sub ID	Mel bins	Accuracy	Log loss	Model size
baseline [18]	40	47.7%	1.473	90.3 KB
1	32	64.4%	1.042	121.3 KB
2	64	68.8%	0.932	121.4 KB
3	128	70.9%	0.859	121.6 KB
4	160	70.5%	0.848	121.8 KB

Our ASC systems have significantly higher performance on the development dataset than the baseline system [18].

4. CONCLUSION

In this work, we have proposed and described a new architecture of convolutional neural networks with mobile inverted bottleneck blocks for acoustic scene classification tasks. Our best ASC system has a very small number of non-zero parameters and has significantly higher performance on the official development dataset than the baseline system. Our best system achieves a classification macro-average accuracy of 70.5%, log loss of 0.848, and has 62,346 non-zero parameters.

5. REFERENCES

[1] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, submitted. [Online]. Available: https://arxiv.org/abs/2005.14623

[2] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[3] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[4] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," *IEEE Signal Processing Letters*, vol. 24, p. 279–283, Mar 2017.

[5] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from Between-class Examples for Deep Sound Recognition," arXiv preprint http://arxiv.org/abs/1711.10282 arXiv:1711.10282, 2017.

[6] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *The 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[7] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," arXiv preprint https://arxiv.org/abs/2104.00298 arXiv:2104.00298, 2021.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[9] S. Gupta and B. Akin, "Accelerator-aware neural network design using automl," arXiv preprint https://arxiv.org/abs/2003.02838 arXiv:2003.02838, 2020.

[10] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[11] W. C. Daniel S. Park, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *INTERSPEECH*, 2019.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[14] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," arXiv preprint https://arxiv.org/abs/1710.05941 arXiv:1710.05941, 2017.

[15] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[16] L. N. Smith and N. Topin, “Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates,” arXiv preprint http://arxiv.org/abs/1708.07120 arXiv:1708.07120.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[18] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, “Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems,” 2021.