DATA EFFICIENT ACOUSTIC SCENE CLASSIFICATION USING SING TEACHER-INFORMED CONFUSING CLASS INSTRUCTION

Technical Report

Jin Jie Sean Yeo[1*], *Ee-Leng Tan*[1], *Jisheng Bai*[2], *Santi Peksi*[1], *Woon-Seng Gan*[1],

[1] Smart Nation TRANS Lab,
50 Nanyang Avenue, Singapore 639798, ye0024an@e.ntu.edu.sg, {etanel, speksi, ewsgan}@ntu.edu.sg
[2] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China,
baijs@mail.nwpu.edu.cn

ABSTRACT

In this technical report, we describe the SNTL-NTU team's submission for Task 1 *Data-Efficient Low-Complexity Acoustic Scene Classification* of the detection and classification of acoustic scenes and events (DCASE) 2024 challenge. Three systems are introduced to tackle training splits of different sizes. For small training splits, we explored reducing the complexity of the provided baseline model by reducing the number of base channels. We introduce data augmentation in the form of *mixup* to increase the diversity of training samples. For the larger training splits, we use *FocusNet* to provide confusing class information to an ensemble of multiple *Patchout faSt Spectrogram Transformer (PaSST)* models and baseline models trained on the original sampling rate of 44.1 kHz. We use Knowledge Distillation to distill the ensemble model to the baseline student model. Training the systems on the *TAU Urban Acoustic Scene 2022 Mobile development dataset* yielded the highest average testing accuracy of (62.21, 59.82, 56.81, 53.03, 47.97)% on split (100, 50, 25, 10, 5)% respectively over the three systems.

Index Terms— Acoustic scene analysis, Depthwise Convolutional Networks, Patchout FaSt Spectrogram Transformer (PaSST), FocusNet, Knowledge Distillation

1. INTRODUCTION

Task 1 of the DCASE 2024 challenge [1] involves acoustic scene classification (ASC) for 10 scenes from 12 cities using 1-sec audio samples. This task includes data-efficient training splits (5, 10, 25, 50, 100)% with constraints on model complexity (128 kB memory, 30 MMACs).

In this submission, three systems are presented. The first system is an adapted baseline model. The second model utilizes a Teacher-Student framework using knowledge distillation [2]. The third system uses the student from the second system as a teacher model for FocusNet [3].

The Baseline (BL) model [4] provided is a simplified version of the CP-Mobile convolutional neural network (CNN) [5, 6]. In this work, we adapt the BL model and find the optimal model complexity for small training splits. We refer to the adapted baseline as the N-Base Channel Baseline (N-BCBL), where N refers to the number of base channels. These optimizations aim to prevent over-fitting on the limited training data for smaller training splits.

CNNs are widely recognized for their state-of-the-art results in the DCASE Task 1 challenge. Numerous CNN architectures have demonstrated significant success in acoustic scene classification (ASC) tasks, particularly on the TAU urban acoustic scene 2022 mobile dataset [7]. The Patchout FaSt Spectrogram Transformer (PaSST) [8] was shown to be an effective teacher for CNNs [5, 9], achieving the top rank in two consecutive DCASE Task 1 challenges.

Despite the state-of-the-art performance of the models, a study of the class-wise accuracy achieved by the top ranking models for DCASE Task 1 over the years reveals that classes such as *street_pedestrian* and *public_square* remain difficult to classify due to the samples sharing acoustic properties of other classes. We introduce *FocusNet* [3] to tackle this issue, by encouraging a student model to pay more attention to these confusion classes.

This report is organized as follows. In section 2, the input features, augmentation techniques used, and model complexity scaling approaches are discussed. Section 3 presents the submitted systems for the challenge, and their training methodology. Section 4 introduces the teacher-student training framework. Section 5 presents the results of our submissions on the development dataset. The report is concluded in Section 6.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Preprocessing

For N-BCBL models, we use 44.1 kHz audio to compute log-mel spectrograms with 256 frequency bins. Short Time Fourier Transformation (STFT) is applied with a window size of 75 ms and a hop size of 11 ms. The model has 35K parameters with 22.6 M MACS.

For PaSST models, we use audio at a sampling rate of 44.1 kHz to compute log-mel spectrograms with 128 frequency bins. STFT is applied with a window size of 18 ms and a hop size of 7 ms. The PaSST model are pre-trained on Audioset and fine-tuning is performed using the log-mel features described above. Pitch shifting is applied by randomly changing the maximum frequency of the Mel filter bank [10].

For student models, 32-BCBL models are trained on audio resampled to 32 kHz to compute log-mel spectrograms with 256 frequency bins. Short Time Fourier Transformation (STFT) is applied

*This research is supported by the Singapore Ministry of Education, Academic Research Fund Tier 2, under research grant MOE-T2EP20221-0014.

with a window size of 96 ms and a hop size of 16 ms. The student models have 61K parameters with 29.4 M MACS.

We remove the random roll augmentation as we did not observe a noticeable improvement in the development set. Frequency masking with a maximum size of 48 Mel bins is applied to all models.

2.2. Frequency-MixStyle Device Impulse Response Augmentation and Mixup

We implement Freq-MixStyle (FMS) [9, 11], which normalizes the frequency bands in a spectrogram and then denormalizes them with mixed frequency statistics of two spectrograms. FMS is randomly applied to each batch with a probability, $p_{FMS} = 0.4$. The mixing coefficient is drawn from a beta distribution parameterized by the hyperparameter α. $\alpha = 0.3$ is used for all experiments.

The device impulse responses (DIRs) from MicIRP [1] were used to augment the waveforms. DIR augmentation was applied to each batch with a probability, $p_{DIR} = 0.6$.

Mixup [12] is designed to improve the generalization ability of deep neural networks by creating new training examples through linear interpolation of existing samples. Mixup generates synthetic samples by combining pairs of training data and their corresponding labels. The mixing coefficient is drawn from a beta distribution parameterized by the hyperparameter α. We use $\alpha = 1$ for all experiments.

2.3. Channel Scaling

The BL model's complexity can be adjusted in 3 ways. Namely, the number of base channels, expansion rate and channel multiplier. In this work, we adjust the base channels while keeping the expansion rate and channel multiplier constant. By reducing the model parameters, we aim to reduce over-fitting of the model on the small training splits.

3. SUBMITTED MODELS

3.1. N-Base Channel Baseline Model

To optimize the model for small training splits, the complexity was reduced by reducing the number of base channels. In a CNN, the channel dimension typically increases while the feature map size decreases. By lowering the initial number of base channels, we reduce the total number of channels throughout the network. This, in turn, decreases the number of parameters, simplifying the model and potentially improving its performance on smaller datasets.

We refer to this approach as the N-Base Channel Baseline (N-BCBL) model. The N-BCBL model is trained using mixup, FMS and DIR augmentation. In this report, we submit the 24-BCBL as one of our submission models as it yields the highest performance for small training splits.

3.2. Knowledge Distillation Ensemble Model

System 2 of our submission is the Knowledge Distillation Ensemble (KD-Ensemble) model. The KD-Ensemble uses knowledge distillation to compress the knowledge of an ensemble of teacher models to a 32-BCBL student. The teacher model ensemble consists of three PaSST and three 32-BCBL models. All 32-BCBL teacher models are trained using mixup. The teacher models are trained in 3 configurations - Freq-MixStyle only, DIR augmentation only or Freq-MixStyle and DIR augmentation.

The use of N-BCBL models in the ensemble is inspired by CPJKU's use of ResNet teacher models in their ensemble, to provide more diverse predictions to the teacher ensemble. The teacher logits are ensembled by averaging the logits of all teacher models. The logits are then used in Knowledge Distillation (KD) training of a N-BCBL student.

For PaSST, the training protocol is the same as in [13]. For the N-BCBL models, mixup is used for all samples, random roll is removed and the number of warm-up steps in the learning rate schedule are reduced to 100.

[1] http://micirp.blogspot.com

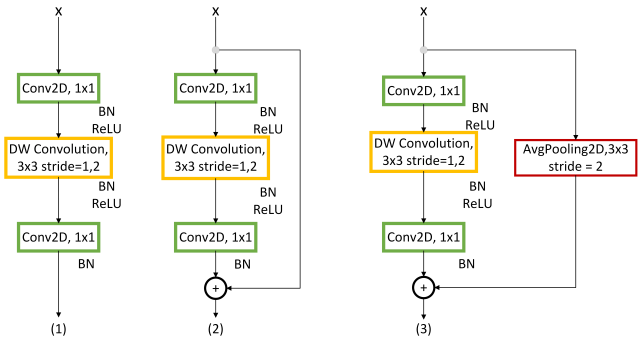


Figure 1: CPM blocks: (1) Transition Block (input channels $\neq$ output channels put channels), (2) Standard Block, (3) Spatial Downsampling Block

3.3. Teacher-Focused Student Model

System 3 of our submission is the Teacher-Focused Student (TFS) model. The student model of system 2 acts as a teacher model and is used to generate pre-computed logits of the training data. FocusNet uses the logits from the teacher model to identify classes that are difficult to classify. This acts as a form of attention, which then encourages the student to learn more discriminative features. FocusNet does not change the architecture of the models and thus does not add to the complexity of the model.

3.4. Model Architecture

The N-BCBL model is an adaptation of the baseline based on the CP-Mobile architecture which uses CPM blocks described in [4] without Global Response Normalization (GRN) after each block. Figure 1 shows the architecture of the CPM blocks used in the BL architecture.

4. TEACHER-STUDENT TRAINING METHODS

4.1. Knowledge Distillation

Knowledge distillation is a model compression technique where a smaller, student model is trained to replicate the behavior of a larger, more complex model teacher model. Equation (1) shows the distillation loss, $\mathcal{L}_{kd}$. It includes both the categorical cross entropy (CCE) loss H and the Kullback-Leibler Divergence loss KL. The parameter λ is used to balance the contributions of the label and distillation losses. Here, z_s and z_t represent the logits from the student

Table 1: 24-BCBL Architecture

Input	Operator	Stride
256 x 89 x 1	Conv2D@3x3, BN, ReLU	2 x 2
128 x 45 x 6	Conv2D@3x3, BN, ReLU	2 x 2
64 x 23 x 24	CPM Block S	1 x 1
64 x 23 x 24	CPM Block D	2 x 2
64 x 23 x 24	CPM Block S	1 x 1
64 x 12 x 24	CPM Block T	2 x 1
32 x 12 x 40	CPM Block S	1 x 1
32 x 12 x 40	CPM Block T	1 x 1
32 x 12 x 80	Conv2D@1x1, BN	1 x 1
32 x 12 x 10	Avg. Pool	-

Input: Frequency Bands x Time Frames x Channels
CPM Block S/D/T: Standard/Downsampling/Transition

Table 2: 32-BCBL Architecture

Input	Operator	Stride
256 x 64 x 1	Conv2D@3x3, BN, ReLU	2 x 2
128 x 33 x 8	Conv2D@3x3, BN, ReLU	2 x 2
64 x 17 x 32	CPM Block S	1 x 1
64 x 17 x 32	CPM Block D	2 x 2
64 x 17 x 32	CPM Block S	1 x 1
64 x 9 x 32	CPM Block T	2 x 1
32 x 9 x 56	CPM Block S	1 x 1
32 x 9 x 56	CPM Block T	1 x 1
16 x 9 x 104	Conv2D@1x1, BN	1 x 1
16 x 9 x 10	Avg. Pool	-

Input: Frequency Bands x Time Frames x Channels
CPM Block S/D/T: Standard/Downsampling/Transition

and teacher networks, respectively. y represents training labels. T is a temperature parameter that softens the probability distributions generated by the softmax activation function δ.

$$\mathcal{L}_{kd} = (1-\alpha)H(\delta(z_s), y) + \alpha \cdot T^2 \cdot KL(\delta(z_t/T), \delta(z_s/T)), \quad (1)$$

4.2. FocusNet

FocusNet aims to reduce misclassification rates by focusing on highly confused classes. The loss functions in FocusNet achieve three objectives: distinguishing between hard and easy samples, focusing on confusing classes, and preventing over-confident predictions by the student model. The overall loss function, $\mathcal{L}_{focus}$, is:

$$\mathcal{L}_{focus} = \mathcal{L}_{\text{cls}} + \alpha\mathcal{R}_{\text{attention}} - \beta\mathcal{R}_{\text{entropy}}, \quad (2)$$

where α and β are hyperparameters between 0 and 1. $\mathcal{L}_{\text{cls}}$ is the classification loss, $\mathcal{R}_{\text{attention}}$ is the attention loss, and $\mathcal{R}_{\text{entropy}}$ is the regularization term. We set α and β to 1 for all TFS models. The classification loss first measures the difference in predictions d_n, between the student and teacher model and is given by

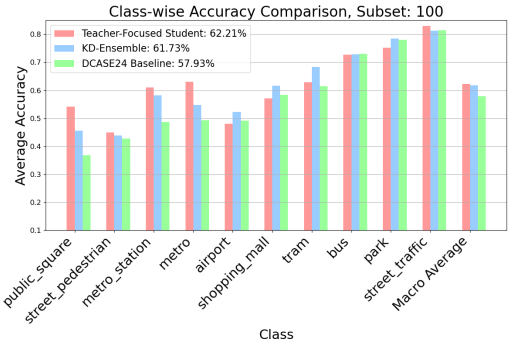


Figure 2: Class-wise Accuracy of Student models and Baseline for the 100% split

$d = \delta(z_s) - \delta(z_t)$. z_s and z_t are the logit values of the student and teacher models, respectively. d_n then modifies prediction $\hat{y}_n$ as given by $\hat{y} = \delta(d + z_s)$.

The classification loss $\mathcal{L}_{\text{cls}}$ is computed as the cross-entropy between the ground truth label y and the modified prediction $\hat{y}$:

$$\mathcal{L}_{\text{cls}} = H(\mathbf{y}, \hat{\mathbf{y}}). \quad (3)$$

Next, a multi-warm label (mwl) mask vector l' is created based on the sign of the individual teacher model's logits z_n^t:

$$l'_n = \begin{cases} 1, & \text{if } z_n^t > 0 \\ 0, & \text{if } z_n^t \leq 0 \end{cases} \quad (4)$$

The mwl vector l'' is formed by adding the mask to the ground truth label y_n:

$$l''_n = \min(1, l'_n + y_n). \quad (5)$$

The normalized mwl vector l is then computed as follows:

$$l_n = \frac{l''_n}{\sum_{j=1}^{N} l''_j}. \quad (6)$$

Finally, the attention loss is calculated as the cross-entropy between l and student predictions:

$$\mathcal{R}_{\text{attention}} = H(l, \delta(z_s)), \quad (7)$$

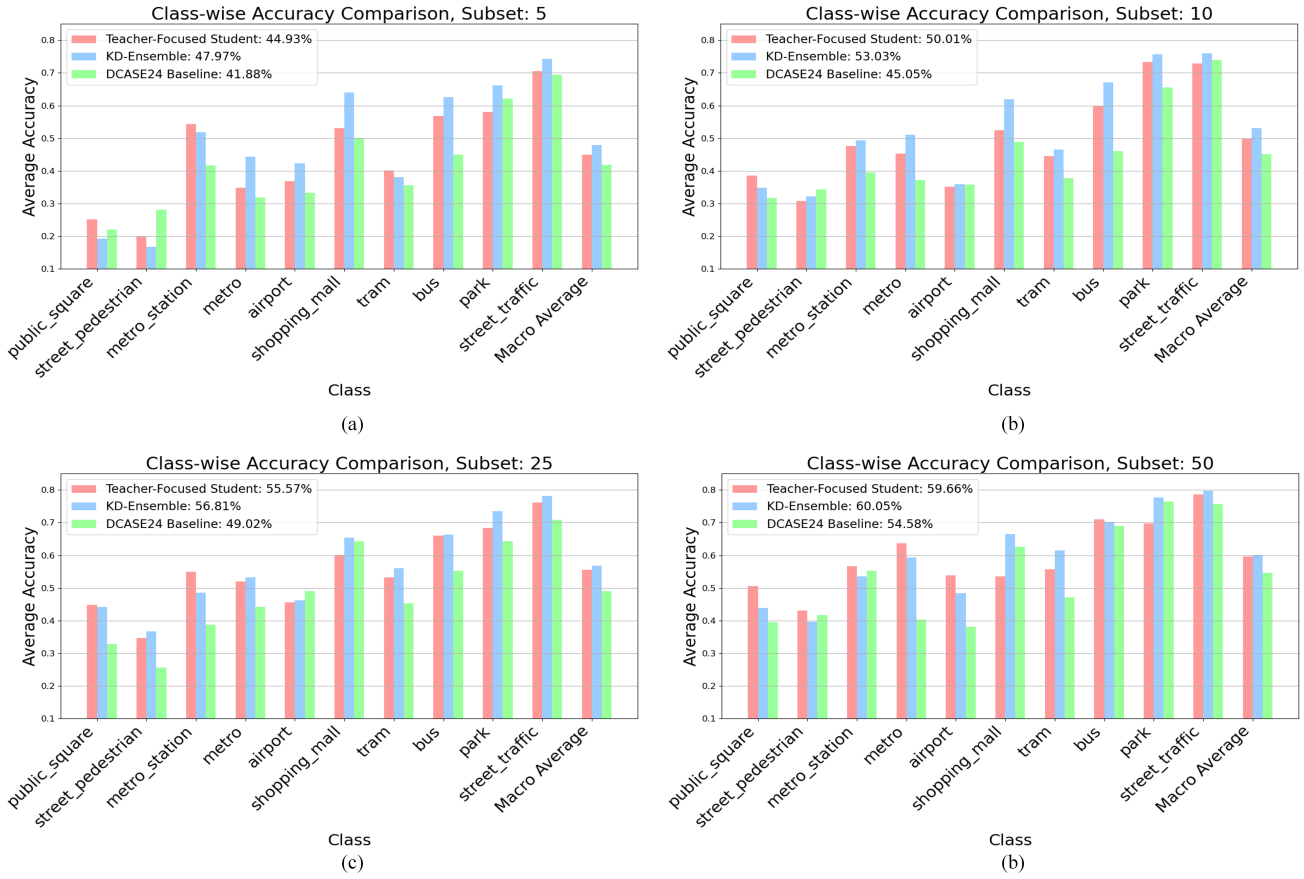
The regularization term $\mathcal{R}_{\text{entropy}}$ encourages the student model to make higher entropy predictions and is computed as follows:

$$\mathcal{R}_{\text{entropy}} = H(\delta(z_s), \delta(z_s)) = -\sum_{n=1}^{N} \delta(z_s) \log(\delta(z_s)). \quad (8)$$

The regularization term is subtracted in (2) to penalize low entropy distributions, promoting higher entropy in student model predictions.

5. RESULTS

Table 3 shows the accuracy results of the three systems. The TFS model achieves the highest accuracy for the 100% split, while the KD-Ensemble achieves the highest accuracy for all other splits.



(a) (b) (c) (b)

Figure 3: Class-wise Accuracy of student models and Baseline on various splits: (a) 5%, (b) 10%, (c) 25%, (d) 50%

Figure 2 illustrates that the accuracy for confusing classes such as *street_pedestrian*, *public_square*, *metro*, and *metro_station* improves over the KD-Ensemble and baseline models. In contrast, easy-to-classify classes like *park*, *bus*, and *tram* show performance degradation against the KD-Ensemble and, in some cases, the baseline.

While the 24-BCBL model outperforms the baseline model (Table 3), both the KD-Ensemble and TFS outperform the 24-BCBL on all splits. This indicates that CNNs can benefit from the guidance of a more complex teacher, even when training data is limited.

For smaller splits, the KD-Ensemble outperforms both the TFS and 24-BCBL models. However, the ability to classify confusing classes remains effective even with small training splits as seen in Fig. 3. An interesting research direction could be to ensemble students trained using KD and TFS models, which we leave as future work.

Table 3: Macro Average Accuracy Comparison

Train split	24-BCBL	KD-Ensemble	TFS	BL Model
5%	43.68	**47.97**	44.93	42.40
10%	47.21	**53.03**	50.01	45.29
25%	53.39	**56.81**	55.57	50.29
50%	55.50	**59.82**	59.66	53.19
100%	57.59	61.74	**62.21**	56.99

6. CONCLUSION

This report described the NTU-SNTL submission to Task 1 of the DCASE 24 challenge. Key contributions include optimizing the N-BCBL model for small datasets by reducing complexity, improving generalizability using mixup augmentation, and training a teacher-focused student by employing knowledge distillation on an ensemble of PaSST and 32-BCBL models. TFS students exhibited a trade-off in classification performance between confusing classes and easy classes. The ASC task has potential for further performance gains where TFS models are ensembled with KD-ensemble students.

7. ACKNOWLEDGMENT

This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20221-0014).

8. REFERENCES

[1] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-Efficient Low-Complexity Acoustic Scene Classification in the DCASE 2024 Challenge," pp. 3–6, 2024. [Online]. Available: http://arxiv.org/abs/2405.10018

[2] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *NIPS Deep Learning and Representation Learning Workshop*, pp. 1–9, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[3] X. Zhang, Z. Sheng, and H. L. Shen, "FocusNet: Classifying better by focusing on confusing classes," *Pattern Recognition*, vol. 129, 2022.

[4] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the Knowledge of Transformers and {CNNs} with {CP}-Mobile," *Detection and Classification of Acoustic Scenes and Events (DCASE)*, no. September, pp. 171–175, 2023.

[5] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," 2022. [Online]. Available: https://github.com/CPJKU/cpjku_

[6] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: efficient acoustic scene classification with cp-Mobile," 2023. [Online]. Available: https://github.com/fschmid56/cpjku_

[7] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions," no. November, 2020. [Online]. Available: http://arxiv.org/abs/2005.14623

[8] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022-Septe, pp. 2753–2757, 10 2021. [Online]. Available: http://arxiv.org/abs/2110.05069http://dx.doi.org/10.21437/Interspeech.2022-227

[9] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge Distillation From Transformers For Low-complexity Acoustic Scene Classification," no. November, pp. 1–5, 2022. [Online]. Available: https://github.com/CPJKU/cpjku_dcase22

[10] K. Koutini, J. Schlüter, and G. Widmer, "Detection and Classification of Acoustic Scenes and Events 2021 Challenge CPJKU SUBMISSION TO DCASE21: Cross-device Audio Scene Classification With Wide Sparse Frequency-damped Cnns Technical Report," 2021. [Online]. Available: https://github.com/kkoutini/

[11] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain Generalization with Relaxed Instance Frequency-wise Normalization for Multi-device Acoustic Scene Classification," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022-Septe, no. Mi, pp. 2393–2397, 2022.

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

[13] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-Robust Acoustic Scene Classification via Impulse Response Augmentation," 2023. [Online]. Available: http://arxiv.org/abs/2305.07499