FMSG-NTU SUBMISSION FOR DCASE 2022 TASK 4 ON SOUND EVENT DETECTION IN DOMESTIC ENVIRONMENTS

Technical Report

Tanmay Khandelwal[1,2]*, Rohan Kumar Das*[1]*, Andrew Koh*[2] *and Eng Siong Chng*[2]

[1]Fortemedia Singapore, Singapore
[2]Nanyang Technological University (NTU), Singapore
f20170106p@alumni.bits-pilani.ac.in, rohankd@fortemedia.com, andr0081@e.ntu.edu.sg, aseschng@ntu.edu.sg

ABSTRACT

In this work, we describe the jointly submitted systems by Fortemedia Singapore (FMSG) and Nanyang Technological University (NTU) for DCASE 2022 Task 4: sound event detection in domestic environments. The proposed framework is divided into two stages: Stage-1 focuses on the audio-tagging system, which assists the sound event detection system in Stage-2. We train the Stage-1 utilizing a strongly labeled set converted into weak predictions, a weakly labeled set, and an unlabeled set to develop an effective audio-tagging system. This audio-tagging system is then used to infer on the unlabeled set to generate reliable pseudo-weak labels, which are used together with the strongly labeled set and weakly labeled set to train the sound event detection system at Stage-2. In Stage-1, we used two different networks, which are frequency dynamic (FDY)-convolutional recurrent neural network (CRNN) and convolutional neural network (CNN)-14 based pretrained audio neural networks (PANNs) for our developed systems. While the system at Stage-2 is based on FDY-CRNN for all the systems submitted to the challenge. It is noted that the systems at both stages employ data augmentation to reduce the risk of overfitting, and apply adaptive post-processing techniques to further enhance the performance. On the DESED real validation dataset, we obtain the highest PSDS1 and PSDS2 of 0.474 and 0.840, respectively.

Index Terms— Sound event detection, semi-supervised learning, CRNN, interpolation consistency training, DCASE 2022.

1. INTRODUCTION

The goal of sound event detection (SED) is to identify the temporal onset and offset and categorize specific sound event types in a variety of sound environments. Its applications include audio surveillance in a variety of environments such as smart homes and cities [1]. The detection and classification of acoustic scenes and events (DCASE) challenge series aim to spearhead the research and developments in SED applications as one of the challenge tasks.

This technical report describes our systems submitted to DCASE 2022 Task 4[1] on sound event detection in domestic environments. This is a follow-up of DCASE 2021 Task 4, whose goal is to recognize sound events inside audio clips utilizing training data from real recordings that are both weakly labeled and unlabeled, as well as synthetic audio clips that are extensively labeled. Additionally, this year, participants are encouraged to leverage external data and pre-trained models to improve the SED systems.

In our submission, we employ two neural networks with the system divided into two stages and multiple strategies as below:

- Frequency dynamic (FDY)-convolutional recurrent neural network (CRNN) [2] and convolutional neural network (CNN)-14 based pretrained audio neural networks (PANNs) [3].
- Interpolation consistency training (ICT) [4] to enhance model robustness.
- Weak training with transformed strong labels into weak labels to improve audio-tagging.
- Infer on the unlabeled set to generate reliable pseudo-weak labels to improve sound event detection performance.

To further improve the performance, we perform:

- Data augmentation methods including mixup [5], frame-shift [6], Gaussian noise addition, filter-augmentation [6] and time-masking [7] to increase data diversity.
- Exponential softmax pooling function [8] to replace the attention pooling in the baseline.
- Asymmetric focal loss [9] to replace binary cross-entropy loss function.
- Adaptive median-filtering for each class to smooth the outputs.

The remainder of the technical report is structured as follows: The dataset used to train and validate the systems is discussed in Section 2. Section 3 details the two-stage architecture, with the training methods used in each stage. Section 4 reports the methods for improving the system's robustness. In Section 5, we described the configuration of each system submitted in the challenge. Finally, Section 6 presents the results of the submitted systems on the DESED real validation set.

2. DATASET

The dataset for the task is primarily based on the DESED [10] dataset, being used since DCASE 2020 Task 4. It is composed of 10 seconds audio clips either taken from AudioSet [11] or synthesized with isolated sound events and backgrounds using Scaper[2] to simulate a domestic environment. The development training set is divided into 3 major subsets:

- 1,578 real recordings with weak annotations.
- 14,412 real recordings, unlabeled in the domain training set

[1]http://dcase.community/challenge2022/

[2]https://github.com/justinsalamon/scaper

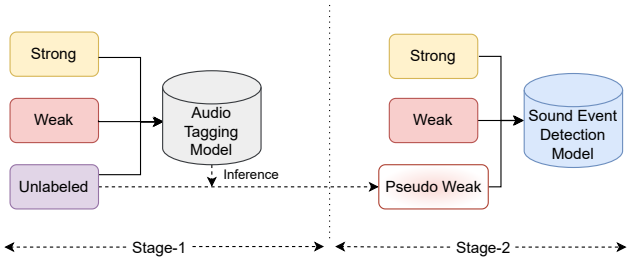

Figure 1: The two-stage learning setup, with Stage-1 focusing on audio-tagging and Stage-2 focusing on sound event detection.

- 10,000 synthetic recordings with strong annotations.
- This year, an additional subset retrieved from the recently released strongly labeled subset of AudioSet consisting of 3,470 real recordings with strong annotations is also released, which is considered as external data.

The development validation set contains, 1,168 real recordings with strong annotations.

3. TWO-STAGE ARCHITECTURE

We adopted a two-stage setup as the basic framework, with Stage-1 focused on audio-tagging and Stage-2 focused on sound event detection [12], as illustrated in Figure 1. The Stage-1 system is used to generate reliable pseudo-weak labels from the unlabeled set that are used by Stage-2 for training. This section further explains the stages in detail.

3.1. Stage-1

In this work, taking inspiration from a prior work [6], which utilizes only the weakly labeled set, we propose a weak training method to have an improved audio-tagging system at Stage-1. We converted the strongly labeled set into a weakly labeled set by removing the onset and offset and keeping the event labels as weak predictions. Then we trained the audio-tagging system using pseudo-weak labels from the strongly labeled synthetic and real set, weakly labeled set, and unlabeled. We used two different architectures in the Stage-1, which are described in the following subsections.

3.1.1. FDY-CRNN

We used frequency dynamic convolution proposed in [2] that applies frequency adaptive kernel in order to enforce frequency dependency on 2D convolution. In the baseline [13] depicted in Figure 2 (a), we replaced the normal convolutional blocks with FDY-convolutional blocks, as illustrated in Figure 2 (b). The CNN feature extractor is composed of 7 layers with each layer having 16, 32, 64, 128, 128, 128, 128 feature maps, respectively, and a kernel size of 3×3. In the FDY-convolutional block, batch normalization [14] is applied after each convolution followed by gated linear unit (GLU) [15] as the non-linear activation function.

3.1.2. CNN-14 based PANNs

We employed CNN-14 based PANNs for pre-trained embeddings as the feature extractor as an alternative to FDY-CRNN in Stage-1. The

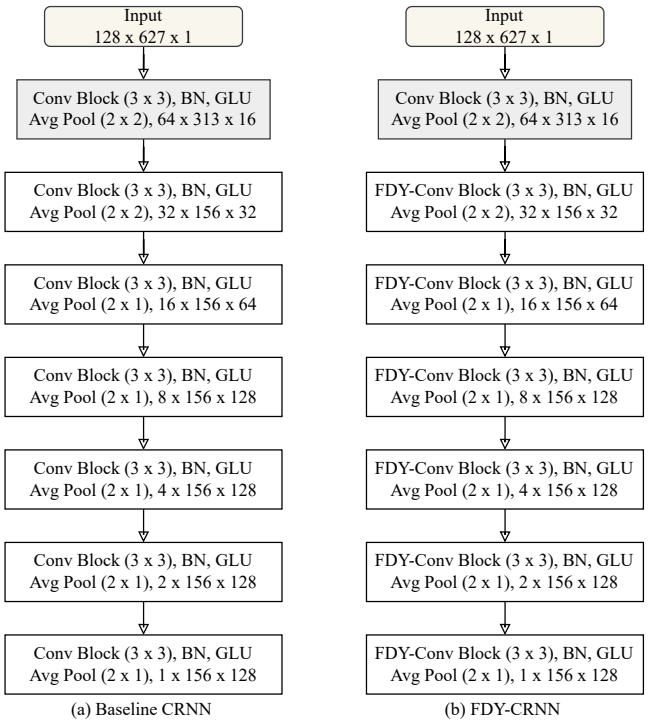

Figure 2: Stage-2 setup (a) CRNN used in baseline (b) FDY-CRNN.

embedding features are used as an input to the bidirectional gated recurrent unit (Bi-GRU). The parameters of the PANNs embeddings are unfrozen and trained. It is noted that the setup is fine-tuned on the DCASE 2022 Task 4 dataset. The 14-layer CNN feature extractor consists of 6 convolutional blocks. Each convolutional block consists of 2 convolutional layers with a kernel size of 3×3. In addition, each convolutional layer is followed by batch normalization [14] and ReLU [16] non-linearity to stabilize the training. Average pooling of 2×2 is applied to each convolutional block for down-sampling. The RNN part following the feature extractor is composed of 2-layers of Bi-GRU with 1024 hidden units. The output of the RNN is followed by a dense layer with sigmoid activation to produce frame-level predictions, and the aforementioned linear layer is multiplied by a dense layer with a softmax activation function to produce clip-level predictions.

3.2. Stage-2

In this work, we used the audio-tagging (Stage-1) based system to make predictions on the unlabeled set to use them as pseudo-weak labels in Stage-2 training, as illustrated in Figure 1. We believe this way we are able to generate reliable pseudo labels, which can help the SED model at Stage-2. In Stage-2, we only used the FDY-CRNN based frequency-dependent architecture described in Section 3.1.1. It is trained on pseudo weakly labeled set, in addition to the strongly labeled set and the weakly labeled set.

4. METHODS

4.1. Semi-Supervised Learning

In this work, we use mean teacher (MT) [17], a semi-supervised learning approach, to learn from unlabeled training data. The mean teacher technique is made up of two network models: the student model and the teacher model. Both of the models have the same architecture, where the student's weights are updated using gradient back propagation, and the teacher's weights are updated as an exponential moving average of the student weights after each epoch.

In addition to MT, we employ another semi-supervised learning method called interpolation consistency training (ICT), which is applied to both the stages [4]. It trains the model to provide consistent predictions at interpolations of unlabeled points (u_j, u_k), as shown below:

$$f_\theta(\lambda u_j + (1-\lambda)u_k) \approx \lambda f'_\theta(u_j) + (1-\lambda) f'_\theta(u_k) \quad (1)$$

where f_θ and f'_θ denote a student model and a teacher model, respectively. The λ is randomly sampled from the Beta distribution [4]. The ICT substitutes all input samples with interpolation samples helping the model to improve the generalization ability, thus our loss function is a sum of the baseline loss function and loss with the interpolation samples as the inputs.

4.2. Data Augmentation

To artificially increase the amount of data and avoid the risk of overfitting, we used several data augmentation techniques during the training in both stages, such as time-masking [7], frame-shifting [6], mixup [5], addition of Gaussian noise and filter augmentation [6]. Time-masking applies weights to the bins of time-frequency representation, whereas frame-shifting shifts the features and labels along the time-axis. Again, mixup randomly mixes selected samples with a mixing parameter, helping in linear interpolation to improve the robustness of the model. In addition, filter augmentation, which uses varying weights on random frequency regions, has been shown to significantly improve SED performance.

4.3. Pooling Function

Taking inspiration from a prior work [8], we employed exponential softmax to replace the attention pooling used in the baseline. The exponential softmax function assigns a weight of $\exp(y_i)$ to the frame-level probability y_i as given below:

$$y = \frac{\sum_i y_i \exp(y_i)}{\sum_i \exp(y_i)} \quad (2)$$

where y_i is the predicted probability of an event occurring in the i^{th} frame. This implies that, with a higher prediction probability, the higher the exponential weight is assigned to the frame-level probability. Hence, it is better under the stringent evaluation criteria for the correctness of the category.

4.4. Asymmetric Focal Loss

Asymmetric focal loss (AFL) [9] function is used to control the training weight depending on the ease and difficulty of the model training. The AFL function for each k^{th} data point with target sound event as y_k and predicted sound event as p_k is given below:

$$l_{AFL}(p, y) = \sum_{n=1}^{K} [(1-p_k)^\gamma y_k \ln p_k + (p_k)^\zeta (1-y_k) \ln(1-p_k)] \quad (3)$$

where the parameters γ and ζ are the weighing hyperparameters given as the input to the function that controls the weight of active and inactivate frames.

4.5. Adaptive Post-Processing

We adopted adaptive post-processing in all the experiments where the median filter window sizes (Win) are different for each event category c calculated based on the varying length of each event in real life as given below:

$$Win_c = duration_c \times \beta_c \quad (4)$$

We took the median duration as $duration_c$ as for some event categories as the variance of the duration is large. Here, we used $\beta = \frac{1}{3}$ and then manually adjusted the window sizes on the validation set. The smoothed result is then converted into binary outputs in the range [0,1] using a threshold of 0.5, as in the baseline.

5. EXPERIMENTAL SETUP

The audio clips are re-sampled at 16kHz to a mono channel using librosa[3]. They are then segmented using a window size of 2048 samples for each subsequent frame with a hop length of 256 samples. The short-time Fourier transform (STFT) is applied on the segmented waveforms to extract their spectrograms. Then log-mel spectrograms are constructed by applying mel-filters in the frequency domain spanning from 0 to 8 kHz, followed by a logarithmic operation. The clips with a duration less than 10 seconds are padded with silence. The batch sizes for all the experiments are [12, 12, 24]. We employed Adam optimizer [18] with a learning rate of 0.001. All the systems described in the further subsections use the interpolation consistency technique, described in Section 4.1. To evaluate the systems, we used polyphonic sound event detection scores (PSDS) [19] on two different scenarios that emphasize different system properties. The system was developed using PyTorch Lightning[4] and trained on NVIDIA Quadro RTX 5000. The following subsections further describe the details of the baselines and our four systems submitted in DCASE 2022 Task 4.

5.1. Baseline-1 (B-1)

The Baseline-1 system given by the organizers utilizes a single-stage CRNN architecture [13]. The CNN part is composed of 7 layers and the RNN part is composed of two layers of bi-GRU with 128 hidden units. The system is trained on synthetic strongly labeled set, weakly labeled set, and unlabeled set using a mean-teacher model approach for 200 epochs.

5.2. Baseline-2 (B-2)

The Baseline-2 system[5] given by the organizers is the additional baseline with the same architecture as that of Baseline-1, utilizing

[3]https://librosa.org/doc/latest/index.html
[4]https://pytorch-lightning.readthedocs.io/en/latest/
[5]https://github.com/DCASE-REPO/DESED_task

AudioSet strongly labeled set in addition to the synthetic strongly labeled set, weakly labeled set, and unlabeled for training.

5.3. Submission-1 (S-1)

We used a two-stage setup in this system submission without the use of any external dataset or pre-trained embeddings, with both stages based on FDY-CRNN trained for 200 epochs each. With the Stage-1 focusing on audio-tagging and Stage-2 on SED. We employed data augmentation methods such as time-masking, frame-shifting, mixup, and addition of Gaussian noise in Stage-1. Additionally, we utilized the AFL function described in Section 4.4 with γ=0.125 and ζ=4 to replace the binary cross-entropy loss function. We also replaced the attention pooling with an exponential softmax function. In Stage-2 based on FDY-CRNN, we used pseudo-weak labels for the unlabeled set generated from Stage-1 using a threshold value of 0.5 in addition to the strongly labeled set, and weakly labeled set. It also employed time-masking, frame-shifting, mixup and filter augmentation for data augmentation, and used the AFL function with γ=0.625 and ζ=1.

5.4. Submission-2 (S-2)

This submission considers only Stage-1 of the two-stage setup to perform audio-tagging and submit the outputs based on it. In this submission, we utilized the CNN-14 based PANNs as pre-trained embeddings instead of the FDY-CRNN architecture used in S-1. The model was trained on weak set, unlabeled set and with synthetic, and real strongly labeled set transformed into weak predictions as indicated in Section 3.1. During inference, we used the exponential softmax function to replace the attention pooling. We utilized time-masking, frame-shifting, mixup, and the addition of Gaussian noise for data augmentation in this setup.

5.5. Submission-3 (S-3)

This submission uses the S-2 as the Stage-1 system of the two-stage setup, since PANNs are well known for audio-tagging. We believe this allows us to generate more reliable pseudo-weak labels for the unlabeled set utilized for training at Stage-2. The Stage-2 of this submission follows the same setup as that in S-1.

5.6. Submission-4 (S-4)

In this submission, we use S-2 as the Stage-1 similar to that in S-3 described in the previous subsection. However, the Stage-2 of this submission has some variations from the system in S-3. We used an exponential softmax function described in Section 4.3 to replace the baseline's attention pooling. In the sigmoid function, we also fine-tuned the temperature parameter to 1.9 instead of its default value of 1. Here, the AFL function is used with γ=0.125 and ζ=4 as the hyperparameters.

6. RESULTS

Table 1 shows the performance of our submitted systems and their comparison to the baselines of DCASE 2022 Task 4 on the real validation set. We observe that our submission S-1 without any external dataset or pre-trained embeddings significantly outperforms the two baselines. This shows the effectiveness of the FDY in CRNN models as well as the two-stage learning setup, which employs weak training with strong labels at Stage-1 in this work. The importance of pre-trained models for SED applications is observed from the usage of CNN-14 based PANNs in submissions S-2, S-3, and S-4. In addition, comparing submission S-2 with S-3, further shows the reliability of the two-stage setup for both aspects of the PSDS metric. While the performance of submission S-4 depicts that the two-stage framework can be adjusted towards aiming an effective audio-tagging, which can be seen from its PSDS2 score.

Table 1: Performance of the baselines and our submitted systems on the real validation set of DCASE 2022 Task 4.

System	PSDS1	PSDS2
B-1	0.336	0.536
B-2	0.351	0.552
S-1	**0.474**	0.730
S-2	0.102	**0.840**
S-3	0.472	0.721
S-4	0.088	0.837

7. ACKNOWLEDGMENT

The authors would like to thank Dr. Teck Kai Chan, a former Fortemedia Singapore employee, for his insightful discussions and suggestions.

8. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[2] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.15296

[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] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3635–3641, 2019.

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017. [Online]. Available: https://arxiv.org/abs/1710.09412

[6] H. Nam, B.-Y. Ko, G.-T. Lee, S.-H. Kim, W.-H. Jung, S.-M. Choi, and Y.-H. Park, "Heavily augmented sound event detection utilizing weak predictions," 2021. [Online]. Available: https://arxiv.org/abs/2107.03649

[7] D. S. Park, W. Chan, 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*, pp. 2613–2617, 2019.

[8] Y. Wang, J. B. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35, 2019.

[9] K. Imoto, S. Mishima, Y. Arai, and R. Kondo, "Impact of sound duration and inactive frames on sound event detection performance," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 860–864, 2021.

[10] N. Turpault, R. Serizel, A. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," *Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, pp. 253–257, 2019.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "AudioSet: An ontology and human-labeled dataset for audio events," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017.

[12] C.-Y. Koh, Y.-S. Chen, Y.-W. Liu, and M. R. Bai, "Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 376–380, 2021.

[13] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 Task 4 technical report," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2019.

[14] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *International Conference on Machine Learning (ICML)*, pp. 933–941, 2017.

[16] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *CoRR*, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[17] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *International Conference on Neural Information Processing Systems*, pp. 1195–1204, 2017.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[19] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 61–65, 2020.