DATA ENGINEERING FOR NOISY STUDENT MODEL IN SOUND EVENT DETECTION

Technical Report

Sangwon Suh, Dong Youn Lee

ReturnZero, Seoul, Korea
{simon, dy}@rtzr.ai

ABSTRACT

This report describes the Sound Event Detection (SED) system for DCASE2022 Task4. We focused on combining data augmentation techniques for the SED mean-teacher system and selecting trainable samples from AudioSet. The neural architecture follows the baseline CRNN model, but a frequency dynamic convolution replaces each convolution layer except the first one. The cost function was also constructed identically to the baseline, but an asymmetric focal loss was used instead of binary cross-entropy for training the AudioSet. The best metrics in the validation set of our experiments were 0.473, 0.723 for PSDS 1 and 2, and 56.9% for color-based F1 scores.

***Index Terms*—** DCASE 2022, sound event detection, data augmentation, mean-teacher, AudioSet, frequency dynamic convolution, asymmetric focal loss

1. INTRODUCTION

In Detection and Classification of Acoustic Scene and Event (DCASE), research on Sound Event Detection (SED) is in progress for classifying and localizing acoustic events. In particular, task 4 aims to detect the activation of ten domestic sound events [1].

The most appropriate dataset for training SED models is strongly labeled data, consisting of class labels with their timestamps. However, annotating the timestamp of sound events is costly and likely to contain many errors, so it is difficult to obtain a sufficient amount of data for training. Therefore, Turpault et al. proposed a heterogeneous dataset that uses weakly labeled and unlabeled data together for training [2]. It was proved that the proposed dataset can further improve the SED mean-teacher model.

In this report, we propose a data augmentation pipeline and a strongly labeled external dataset from AudioSet to enhance supervised training on strong data.

2. BACKGROUNDS

2.1. Frequency dynamic convolution

The frequency dynamic (FDY) convolution is designed to obtain important features in different frequency domains [3]. It consists of multiple kernel basis and their attention weights. The weighted sum of basis kernels implies the weighted sum of features found in the spectrogram with frequency-wise attention. All the systems we submitted have a neural architecture composed of FDY convolutions.

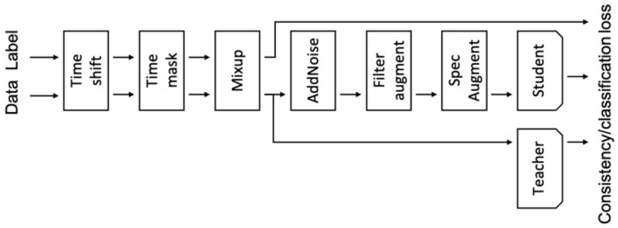

Figure 1: Data augmentation pipeline overview

2.2. Asymmetric focal loss

Asymmetric Focal Loss (AFL) has been proposed to solve the data imbalance problem between active and inactive frames in multi-class SED [4]. Compared to the long-duration events (e.g., electric shaver and vacuum cleaner) and inactive durations, the short-duration events (e.g., cat, dishes, and dog) have complex patterns but less active frames, making it difficult for the model to learn those features. Therefore, the idea is to calculate the focal loss [5] separately weighted for active and inactive, respectively. The following equation describes the AFL:

$$AFL(p) = (1-p)^{\gamma} y log(p) + p^{\zeta}(1-y)\log(1-p), \quad (1)$$

where p is the output probability, y is ground-truth, γ is the focusing parameter for active samples, and ζ for inactive samples. We replaced binary cross-entropy (BCE) with AFL as a supervised loss in training system 3, where the additional data was included for training.

3. DATA AUGMENTATION PIPELINE

We constructed a data augmentation pipeline to effectively train a limited amount of strong labeled data, as shown in Figure 1. Furthermore, to maximize the training effect of the mean-teacher, we added more noise augmentations to the student model input. The following subsections contain details of each data augmentation technique.

3.1. Time shifting and time masking

The intention of utilizing time shifting and time masking is to avoid unintended memorization by temporal position. A few frames of data and labels were shifted and masked randomly.

3.2. Mixup

Mixup is a technique to add-up two samples and labels in a specific ratio [6]. This was initially invented for image classification but is also viable in the audio domain. By adding two samples, the result becomes one sample with two sound events. Furthermore, mixup generates more non-zero frames for training by overlapping inactive frames with other active frames.

3.3. Add noise

Musan is a dataset including various noise samples and consists of speech, music, and noise categories by its source. Thus, many speech domains are using it for adding noise to training samples [7]. However, it is undesirable to utilize the speech and noise categories considering the interference with the feature learning of the targeting classes of DESED. We randomly added music noises to sample while training systems, except for the one system that trained only with internal sets. The random noise was added in case of internal set training.

3.4. FilterAugment and SpecAugment

FilterAugment randomly increases or decreases the frequency band energy of the spectrogram to mimic different acoustic environments. According to Nam et al., this acoustic environment refers to the physical objects surrounding the sound source, various types of receivers, and the air as the medium of waves [8][9]. We applied a linear type of FilterAugment to obtain more natural augmentation results.

SpecAugment [10] located at the end of the pipeline was introduced for frequency masking. Since the regularization effect is sufficient on the temporal axis, the time mask and time stretch were not activated.

4. AUDIOSET DATA SELECTION

AudioSet is an ontology and a dataset for numerous acoustic events. The provided annotations include both weak and strong labels [11][12]. We were interested in the data split named "train_strong" to supplement the strong labeled data. Our intuitions for utilizing AudioSet were: (1) both datasets are sourced from YouTube, (2) both have classes with the same labels (e.g., Blender, Cat, Dog, Electric shaver, Vacuum cleaner, etc.), and (3) some classes in DESED group several classes from AudioSet (e.g., Bow-wow and Bark are grouped to the Dog, etc.)

Since it is not clear whether the class definitions of the two datasets are the same, we tried to find useful samples from AudioSet by examining the prediction scores of the pretrained model. For each class, frame-level prediction scores were micro-averaged to calculate confidence scores for corresponding DESED classes. When the Top-1 confidence score exceeded 0.3, the class is relabeled with the corresponding DESED label and included in the trainable data. After pruning some unreliable and misclassified classes by hand, we selected 500 samples per class to avoid class imbalance. In conclusion, 4,395 samples were selected as trainable data, and the number of samples per class is shown in Table 1.

Table 1: Number of selected AudioSet samples

Class	samples
Alm	500
Bld	373
Cat	500
Dish	500
Dog	500
Shv	369
Fry	500
Wtr	359
Spch	500
Vcm	294
Total	4395

5. EXPERIMENT METHODS

5.1. Datasets and feature extraction

The official development set of DCASE2022 provided three heterogeneous datasets for training: 10,000 of strongly labeled, 1,578 weakly labeled, and 14,412 unlabeled in-domain data. Both weakly and unlabeled data were real audio, but strongly labeled one was synthesized data with Scaper [13]. We used whole official sets for training all systems. For the training system2, we included additional 3,470 clips with strong labels as external data. These data were real-world audio and were provided with the baseline system[1]. In the case of training system 3, we introduced 4,395 additional data from section 4, which was selected from AudioSet. Finally, we included a validation set for training the submission version of each system.

We prepared samples as mono audio with a 16 kHz sampling frequency for our experiments. Each audio was converted to a spectrogram by skipping every 256 samples with a 2048-length window, and each spectrum was aggregated into 128 Mel frequency bins. The extracted filter banks were converted to the log scale and applied min-max scaling.

5.2. Neural architecture

The neural architecture in this experiment follows the baseline system in its overall structure. The CNN structure of the model consists of 7 convolution blocks, and the number of filters is [32, 64, 128, 256, 256, 256, 256], respectively. The first block has a basic 2D convolution layer, followed by six blocks of FDY convolution. Additionally, all convolution blocks use a weighted sigmoid gate [14] as an activation layer and a dropout rate of 0.5. The average pooling layer is used for temporal and frequency pooling, and the size is [[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]], respectively.

[1] https://github.com/DCASE-REPO/DESED_task/tree/master/recipes/dcase2022_task4_baseline

The RNN structure is two layers of 128 bidirectional gated recurrent units. The linear layer follows the RNN with sigmoid activation for strong predictions. The output for weak prediction uses a linear softmax to aggregate the strong predictions with class-wise weighting [15].

5.3. Optimization

The model was trained with Adam optimizer with a maximum learning rate of 0.001. We applied 50 epochs of the ramp-up and 450 epochs of decay strategy to the learning rate scheduler.

For the supervised loss, we used BCE for both weak and strong predictions; but we used the AFL instead of BCE for system 3. We introduced a weighting factor of 0.1 to weak BCE loss to reduce the impact of gradients produced by weak predictions. We used a mean-squared error (MSE) as the consistency loss for weak and strong predictions.

5.4. Post-processing

We applied a temperature to the sigmoid function in inference phase. Zheng et al. claimed that applying the temperature parameter T to the sigmoid function could reduce the polarized distribution of the prediction result [16]. Therefore, we used a temperature of $T = 3$ for the inference process, which led to a slight performance improvement in both PSDS scenarios 1 and 2.

Additionally, we applied different lengths of median filters per event to smooth the prediction results. Delphin-Poulat et al. claimed that the duration of the different sound events varies from one event to another [17]. Accordingly, we divided into two groups, long-duration and short-duration events, by a mean duration of 3 seconds. We tuned median window length towards maximizing PSDS1 applying different search spaces to each group.

For system 4, we applied a specific post-processing method, called weak SED. System 4 is not a sole model but is made from the predictions of System 3 with the weak SED processing. This method, proposed by Nam et al., has a strong positive effect on the PSDS2 by setting timestamps of the weak prediction to the entire duration of audio [8]. Therefore, it can be helpful when predicting sound events whose localization is less important.

6. RESULTS

The in-lab results for the proposed systems on the official validation dataset are shown in Table 2. The submitted version of each system includes the validation set as training data, and the prediction results are ensembles of six random checkpoints between 250 and 500 epochs.

Table 2: Metric results for the official validation set

System	PSDS1	PSDS2	CB-F1
Baseline	0.336	0.536	40.1
Baseline (AudioSet strong)	0.351	0.552	42.9
Baseline (AST)	0.313	0.722	37.2
Zheng et al. [16]	0.454	0.671	52.4
FDY-SED [3]	0.452	0.67	53.3
System1	0.424	0.649	51.0
System2	**0.473**	0.723	**56.9**
System3	0.445	0.704	54.5
System4	0.063	**0.814**	19.3

7. REFERENCES

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

[2] N. Turpault, R. Serizel, "Training sound event detection on a heterogeneous dataset," *arXiv preprint arXiv:2007.03931,* 2020.

[3] H. Nam, et al., "Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection," *arXiv preprint arXiv:2203.15296*, 2022.

[4] K. Imoto, et al. "Impact of sound duration and inactive frames on sound event detection performance," *in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE,* 2021. p. 860-864.

[5] T. Y. Lin, et al. "Focal loss for dense object detection," *in: Proceedings of the IEEE international conference on computer vision.* 2017. p. 2980-2988.

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412,* 2017.

[7] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484,* 2015.

[8] 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," *Tech. Rep., DCASE Challenge,* 2021.

[9] H. Nam, S. H. KIM, and Y. H. Park, "Filteraugment: An acoustic environmental data augmentation method," *in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE,* 2022. p. 4308-4312.

[10] D. S. Park, W., Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *in Proc. Interspeech,* 2019, pp. 2613–2617

[11] J. F. Gemmeke, et al. "Audio set: An ontology and human-labeled dataset for audio events," *in: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE,* 2017. pp. 776-780.

[12] S. Hershey, et al. "The benefit of temporally-strong labels in audio event classification," *in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE,* 2021. pp. 366-370.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," *in Proc. WASPAA, 2017*, pp. 344–348.

[14] M. Tanaka, "Weighted sigmoid gate unit for an activation function of deep neural network," *Pattern Recognition Letters,* 2020, 135: 354-359.

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

[16] X. Zheng, H. Chen, Y. Song, "Zheng ustc team's submission for dcase2021 task4–semi-supervised sound event detection," *Tech. Rep., DCASE2021 Challenge,* 2021.

[17] L. Delphin-Poulat, and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," *Tech. Rep., DCASE2019 Challenge,* 2019.