PRETRAINED MODELS IN SOUND EVENT DETECTION FOR DCASE 2022 CHALLENGE TASK4

Technical Report

Shengchang Xiao, Xueshuai Zhang, Pengyuan Zhang

University of Chinese Academy of Sciences
department of Electronic Engineering
Beijing, China
xiaoshengchang@hccl.ioa.ac.cn

ABSTRACT

In this technical report, we describe our submitted systems for dcase 2022 Challenge Task4: Sound Event Detection in Domestic Environments. Specifically, we submit two different systems respectively for PSDS1 and PSDS2. As PSDS2 focuses on avoiding confusion between classes rather than the localization of sound events, we only predict weak labels of clips to improve PSDS2. Moreover, we apply the pretrained neural networks including PANNs and SSAST in our systems to improve the generalization and robustness of our models. These pretrained models trained on large-scale datasets such as audioset can effectively alleviate the problems of lack of real training data. We fuse multiple pretrained models to make full use of the information of external data, which significantly improve the performance of our systems. In addition, we use various data augmentation techniques to expand provided data. According to the character of each sound event, we use the class-wise median filter and further classify some confusing events. As a result, we achieve the best PSDS1 of of 0.481 and best PSDS2 of 0.826 on the DESED real validation dataset.

Index Terms— Sound Event detection, Semi-supervised learning, pretrained models

1. METHODS

1.1. Dataset

All of the dataset we use in our training are described as follows [1]:

- Unlabeled in domain training set: 14412 clips
- Synthetic strongly labeled training set: 10000 clips
- Synthetic strongly labeled validation set: 2500 clips
- Weakly labeled training set: 1200 clips
- Weakly labeled validation set: 378 clips
- Strongly labeled test set: 1168 clips
- AudioSet strong training set: 2500 clips
- AudioSet strong validation set: 970 clips

We don't use the audioset strong data in base system. In other systems, all of the datasets are used. In addition, we calculate the duration length and occurrences of ten event classes in strongly labeled set and audioset strong set. The result of 4638 clips is shown in Table 1.

Table 1: Duration Length and Occurrences of Ten Event Classes

	Mean	**Mid**	**Occurrences**
Alarm_bell_ringing	2.14	1.03	2143
Blender	5.25	4.80	313
Cat	1.1	0.74	781
Dishes	0.55	0.33	2576
Dog	1.00	0.56	1949
Electric_shaver_toothbrush	7.05	8.96	279
Frying	8.23	10	620
Running_water	6.11	6.09	833
Speech	1.59	1.04	9998
Vacuum_Cleaner	7.86	9.97	178

It's shown that ten event classes can be roughly divided into two groups: long duration and short duration. The long duration group includes Blender, Electric_shaver_toothbrush, Frying, Running_water and Vacuum_Cleaner, which generally last more than 5 seconds and occur less than 900 times. The short duration groups includes Alarm_bell_ringing, Cat, Dishes, Dog, Speech, which generally last less than 2 seconds and occur more than 1000 times.

1.2. Weak Prediction

As PSDS2 focuses on avoiding confusion between classes rather than the localization of sound events, we only predict weak labels of clips and set timestamp to start and end of the entire duration of the audio [2]. Because of the low Detection Tolerance criterion (DTC) [3], this method can greatly improve the PSDS2 scores. Specifically, the duration length of audio is 10 seconds and the parameter of Detection Tolerance criterion is 0.1, which means that as long as the ground true length of event is longer than 1 seconds, the system will regard it as true positive. From the Table 1, we can see that the mean duration length of all events except Dishes is longer than 1 seconds. In reality, there are commonly multiple Dishes events in a clip. If the sum of their duration length is more than 1 seconds, all the Dishes in the clip will be considered as True Positive. Thus the weak prediction can work well in improving PSDS2.

During training stage, we don't use the strongly labeled data. Instead, all the strongly labeled datasets are relabeled weakly including systhetic set and audioset strong set. The loss is calculated as the sum of supervised weak loss and self-supervised weak loss.

The validation metric is defined as F1 score for weak labels. In fact, this method is similar to the audio tagging task. The difference is that we need to set the timestamp equal to the entire duration of clips during inferring stage. In the aspect of model, we adopt the CRNN [4] and pretrained PANNs [5]. Compared to the baseline, we use the output of first fully connected layer as global embedding. Then we fuse the embedding with the output of rnn layer of CRNN. It's worth noting that the parameters in PANNs are not freezed and we train the model end to end.

1.3. Pretrained Models

For PSDS1, we apply the pretrained neural networks including PANNs and SSAST [6] in our systems to improve the generalization and robustness of our models. These pretrained models trained on large-scale datasets such as audioset can effectively alleviate the problems of lack of real training data. We fuse multiple pretrained models to make full use of the information of external data, which significantly improve the performance of our systems.

In particular, the CNN channels are increased form {16, 32, 64, 128, 128, 128, 128} to {32, 64, 128, 256, 256, 256, 256} while the number of RNN cells is increased from 128 to 256. The context gating is used as activation function. And the attention pooling layer is set to time-dimension for aggregating the detection output into audio tagging output. Moreover, we adopt pretrained CNN_16k from PANNs. The output of third cnn layer is extracted as frame-embedding. The self-supervised audio spectrogram transformer(SSAST) is also used as another frame-embedding. Then the frame-embeddings are fed into two different RNNs. The outputs are fused with CRNN. The parameters in PANNs and SSAST are not freezed as well.

1.4. Data augmentation and Post-processing

In our system, we utilize various data augmentation techniques including specaugment [7], mixup [8], frame shift, , FilterAug [2] and add background noise to expand provided data. For specaugment, we apply frequency masking and time masking. The mixup and frame shift strategies is used to enhance the generalization ability. FilterAugment is proposed to consider various acoustic conditions and simulate them. It splits the whole frequency range into several frequency bands and multiplies random factors to these bands. The background noise includes Gaussian white noise, pure music and other free sounds.

Because each event class differs in duration length, we use the class-wise median filter. For each sound event, we search for the optimal median filter length. In addition, we find that some event classes are similar in spectrum and easily confused including Blender and Vacuum_Cleaner, Frying and Running_water. Therefore, we train a new model to classify these classes and set thresholds based on their occurrence frequency.

2. EXPERIMENTS

2.1. Feature Extraction

In our system, we use the same log-mel spectrograms as baseline. The spectrograms are extracted on 16 kHz audio with 128 mel frequency bins, 2048 window length and 256 hop length. As a result, each 10-second audio clip is transformed into a 2D time-frequency representation with a size of (626×128).

2.2. Experimental results

First, we evaluate the data augmentation and post-processing in base system [9] . Our base system consists of CRNN network only and is not using external data for training. The result is shown in Table 2. It can be observed that data augmentation and post-processing can effectively promote the performance with PSDS1 increasing from 0.405 to 0.431 and PSDS2 increasing form 0.611 to 0.645.

Table 2: Results of Data Augmentation and Post-processing

Model	PSDS1	PSDS2
Base	0.399	0.601
Base+data augmentation	0.418	0.626
Base+post-processing	0.408	0.617
Base+data augmentation+post-processing	0.423	0.632

Then we evaluate the performance for weak prediction. This system is fusion model composed of a CRNN network and a pretrained CNN_16k from PANNs. In order to demonstrate the superiority of fusion model, we compare the single model with fusion model based on weak prediction. The result is shown in Table 3. It can be seen that weak prediction contributes to much higher PSDS2 from 0.601 to 0.752 despite the decrease of PSDS1. The fine-tuned CNN_16k also achieves good results. More importantly, the fusion model consisting of CRNN and CNN_16k increases PSDS2 to 0.809.

Table 3: Results of Weak Predictions

Model	PSDS1	PSDS2
CRNN	0.061	0.752
CNN_16k	0.052	0.783
CRNN+CNN_16k	0.058	0.809

Moreover, we eveluate the performance of pretrained models in increasing PSDS1. PANNs and SSAST are both fused into our system. The result is shown in Table 4. All of the models are trained with external dataset. PANNs and SSAST are fine-tuned with provided data. As we can see, the fused model with CNN_16k and SSAST can effectively improve both PSDS1 and PSDS2 scores. The fused model can achieve PSDS1 of 0.459 and PSDS2 of 0.672.

Table 4: Results of Pretrained Models

Model	PSDS1	PSDS2
CRNN	0.424	0.623
CRNN+CNN_16k	0.445	0.650
CRNN+CNN_16k+SSAST	0.459	0.672

Finally, we adopt model ensemble methods to reduce model variance and improve accuracy. The inferred prediction probabilities are averaged of all the models. The ensembled results on shown in Table 5. The base system is the CRNN trained without external dataset. The weak prediction system aims to obtain much higher PSDS1. The pretrained systems have better performance with PSDS2.

Table 5: Results of Ensembled Models

Model	PSDS1	PSDS2
base	0.431	0.645
weak prediction	0.051	0.826
pretrained model1	0.475	0.688
pretrained model2	0.481	0.694

3. CONCLUSION

In this technical report, we describe our submitted systems for dcase 2022 Challenge Task4. We mainly use weak prediction, pretrained models and data augmentation and post-processing to improve PSDS1 and PSDS2 scores. We achieve the best PSDS1 of of 0.481 and best PSDS2 of 0.826 on the DESED real validation dataset.

4. REFERENCES

[1] https://dcase.community/challenge2022/task-sound-event-detection-in-domestic-environments.

[2] 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," *arXiv preprint arXiv:2107.03649*, 2021.

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

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

[5] 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.

[6] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," *arXiv preprint arXiv:2110.09784*, vol. 4, 2021.

[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," *arXiv preprint arXiv:1904.08779*, 2019.

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

[9] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.