GUIDED MULTI-BRANCH LEARNING SYSTEMS FOR SOUND EVENT DETECTION WITH SOUND SEPARATION

Yuxin Huang[1,2], *Liwei Lin*[1,2], *Shuo Ma*[1,2], *Xiangdong Wang*[1*], *Hong Liu*[1], *Yueliang Qian*[1],
Min Liu[3], *Kazushige Ouchi*[3]

[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China,
{huangyuxin18g, linliwei17g}@ict.ac.cn, ms_hebut@163.com, {xdwang, hliu, ylqian}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Toshiba China R&D Center, Beijing, China,
liumin@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp
*Corresponding author

ABSTRACT

In this paper, we describe in detail our systems for DCASE 2020 Task 4. The systems are based on the 1st-place system of DCASE 2019 Task 4, which adopts weakly-supervised framework with an attention-based embedding-level pooling module and a semi-supervised learning approach named guided learning. This year, we incorporate multi-branch learning (MBL) into the original system to further improve its performance. MBL uses different branches with different pooling strategies (including instance-level and embedding-level strategies) and different pooling modules (including attention pooling, global max pooling or global average pooling modules), which share the same feature encoder of the model. Therefore, multiple branches pursuing different purposes and focusing on different characteristics of the data can help the feature encoder model the feature space better and avoid overfitting. To better exploit the strongly-labeled synthetic data, inspired by multi-task learning, we also employ a sound event detection branch. To combine sound separation (SS) with sound event detection (SED), we fuse the results of SED systems with SS-SED systems which are trained using separated sound output by an SS system. The experimental results prove that MBL can improve the model performance and using SS has great potential to improve the performance of SED ensemble system.

Index Terms— Sound event detection, multi-task learning, weakly-supervised learning, sound separation, semi-supervised learning

1. INTRODUCTION

DCASE 2020 task 4 [1] is the follow-up to DCASE 2019 task 4 [2]. While DCASE 2019 task 4 targets on exploring the usage of weakly labeled data, unlabeled data and synthetic data in sound event detection (SED), DCASE 2020 task 4 encourages participants to combine sound separation with SED in addition to the same task in DCASE 2019. There are three subtasks in DCASE 2020 task 4: SED without sound separation, SED with sound separation and sound separation (using the SED baseline system). We participated in the first two subtasks. However, for the second subtask, we just use the baseline system for sound separation provided by the challenge organizer and focus on combination of sound separation and SED.

In this paper, we describe in detail our systems for the two subtasks we participated in DCASE2020 task 4. The systems are based on the first-place system of DCASE 2019 task 4 developed by Institute of Computing Technology (ICT), Chinese Academy of Sciences [3], which adopts the multiple instance learning framework with embedding-level attention pooling [4] and a semi-supervised learning approach called guided learning [5]. The multi-branch learning approach (MBL) [6] is then incorporated into the system to further improve the performance. Multiple branches with different pooling strategies (embedding-level or instance-level) and different pooling modules (attention pooling, global max pooling or global average pooling) are used and share the same feature encoder. To better exploit the synthetic data with strong labels, inspired by multi-task learning [7], a sound event detection branch is also added. Therefore, multiple branches pursuing different purposes and focusing on different characteristics of the data can help the feature encoder model the feature space better and avoid overfitting. To incorporate sound separation into SED, we train models using output of the baseline system of sound separation and fuse the event detection results of models with or without sound separation.

2. THE DCASE 2019 TASK 4 SYSTEM BY ICT

Our systems for DCASE 2020 task 4 follows the framework of the DCASE 2019 task 4 system by ICT [3], which won the 1st place and the reproducible system award in the DCASE 2019 task 4 challenge. The system utilizes convolutional neural network (CNN) with embedding-level attention pooling module for weakly-supervised SED and uses disentangled features to solve the problem of unbalanced data with co-occurrences of sound events [4]. To better use the unlabeled data jointly with weakly-labeled data, the system adopts a semi-supervised learning method named guided learning [5], which uses different models for the teacher model and student model to achieve different purposes implied in weakly-supervised SED. For the synthetic data, the system regards them as weakly annotated training data and the time stamps of sound events in the strong labels are not used. The system is trained by the DCASE 2019 training data, including weakly-labeled data, synthetic data and unlabeled data without data augmentation. The system that won the 1st place in DCASE 2019 task 4 was the ensemble system of 6 systems with the same model architecture, and the ensemble method is averaging all the probabilities output by the systems.

3. METHOD

3.1. Guided learning for semi-supervised SED

We use the guided learning method as our basic model framework. The guided learning method is composed of two parts: a professional teacher model (PT-model) and a promising student model (PS-model). The PT-model is designed to achieve reliable audio tagging. As a result, The instance-level feature generated by the PT-model has large receptive field.

The PS-model is designed to detect the sound events, where the audio tags and event boundaries both need to be predicted. Since the PS-model focuses on frame-level pediction, the instance-level feature generated by the PS-model has small receptive field.

During training, the PT-model and PS-model use the same input data. For the data with weak labels in a batch of input data, the PT-model and PS-model both use the labels as their training target. For event category c, the loss function is calculated as:

$$Loss_{\mathrm{labeled}} = \sum_{c} cross\_entropy(y_c, \hat{\mathbf{P}}(y_c|\mathbf{x})) \tag{1}$$

where y_c is the ground truth.

For the unlabeled data, the PS-model uses the pseudo labels generated by the PT-model as the training target and the PT-model does not have any training target. The pseudo label generated by the PT-model is obtained as:

$$\psi_c^{\mathrm{PT}} = \begin{cases} 1, & \hat{\mathbf{P}}^{\mathrm{PT}}(y_c|\mathbf{x}) \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\hat{\mathbf{P}}^{\mathrm{PT}}(y_c|\mathbf{x})$ denotes the probability of audio tagging output by the PT-model. Then the loss function of the unlabeled data is:

$$Loss_{\mathrm{unlabeled}}^{\mathrm{PS}} = \sum_{c} cross\_entropy(\psi_c^{\mathrm{PT}}, \hat{\mathbf{P}}^{\mathrm{PS}}(y_c|\mathbf{x})) \tag{3}$$

where $\hat{\mathbf{P}}^{\mathrm{PS}}(y_c|\mathbf{x})$ denotes the probability of audio tagging output by the PS-model. After s training epochs, the PS-model is able to achieve reliable audio tagging. Then, the audio tagging pseudo labels of unlabeled data output by the PS-model is also used as the training target of the PT-model. The loss function is calculated as:

$$Loss_{\mathrm{unlabeled}}^{\mathrm{PT}} = \alpha \sum_{c} cross\_entropy(\psi_c^{\mathrm{PS}}, \hat{\mathbf{P}}^{\mathrm{PT}}(y_c|\mathbf{x})) \tag{4}$$

where α is the hyperparameter to adjust the loss weight. In our experiments, we set $\alpha = 1 - 0.997^{epoch-s}$, $s = 15$.

3.2. Multi-branch learning for semi-supervised SED

To further improve the performance, the MBL [6] is incorporated into the guided learning system (MBL-GL). Multiple branches with different pooling strategies such as embedding-level pooling and instance-level pooling and different pooling modules such as attention pooling (ATP), global max pooling (GMP) and global average pooling (GAP), are used and share the same feature encoder. As shown in Figure 1, one branch is set as the main branch which takes part in training and detection and another branch is set as the auxiliary branch which is only used for training. In our system, we apply the MBL into the PS-model. We choose the embedding-level ATP as the main branch and instance-level GMP or instance-level GAP as the auxiliary branch. The loss function is calculated as:

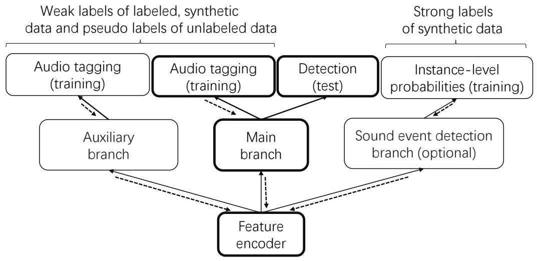


Figure 1: An overview of the architecture of the PS-model

$$Loss_{\mathrm{PS-total}} = aL_{\mathrm{PS-main}} + bLoss_{\mathrm{PS-auxiliary}} \tag{5}$$

where a and b are the loss weights of the main branch and the auxiliary branch. The a is set to 1.0 and the b is set to 0.5 or 1.0 based on the performance of the validation set.

The reason why we apply the MBL method in the PS-model is that the PS-model outputs the final results of SED while the PT-model only outputs the audio tagging results which is only used in the training process of PS-model. In our early study, we found that the improvement of MBL for audio tagging was limited compared with the improvement for SED.

By using multiple branches, we can also fuse the results of both branches to obtain better result at the inference stage. In this paper, if the auxiliary branch is instance-level GAP, we ensemble the detection results of the main branch and auxiliary branch by taking the average results of instance-level probabilities.

$$\hat{\mathbf{P}}_{\mathrm{fusion}}(y_{ct}|\mathbf{x}_t) = \alpha\hat{\mathbf{P}}_{\mathrm{GAP}}(y_{ct}|\mathbf{x}_t) + (1-\alpha)\hat{\mathbf{P}}_{\mathrm{ATP}}(y_{ct}|\mathbf{x}_t) \tag{6}$$

We set $\alpha = 0.5$ in our experiments.

In [6], the MBL approach is proposed only for weakly-labeled data. However, in this work, we need to use the unlabeled data to train our model. In our early experiments, we found that if the ratio between the weakly-labeled data and unlabeled data in each mini-batch was set to be the ratio in the whole training set, which was about 1:9, the MBL-GL performed poorly. The reason for this phenomenon may be that for MBL, although more branches can make the common feature be fit for various learning purposes so that reduce the risk of overfitting, more branches can increase the risk that the common feature can not fit for any learning purpose when the training data contain much noise. For the guided learning framework, the training targets of the unlabeled data for PS-model are produced by the PT-model, which contain noise. To reduce the risk mentioned above, we increase the ratio between the amount of labeled data and unlabeled data to reduce the influence of noise in training data. Besides, different from [6], we only use one auxiliary branch since we find that using two auxiliary branches can decrease the model performance of MBL-GL for the same reason that in guided learning, the training data for the PS-model contain noise and may increase the difficulty to train multiple branches.

3.3. The detection branch for synthetic data

In previous study, the multi-task learning of SED in which detecting sound event boundaries and deciding the existence of sound events are considered as two tasks are proved to be a good method to improve the performance of SED. However, this method needs data with strong labels for training. In this work, only the synthetic data has strong labels. To better exploit the synthetic data with strong labels, inspired by multi-task learning, a sound event detection branch (SEDB) is also added. As shown in Figure 1, only the synthetic data are used for training the SEDB and the output of the SEDB is the probability of each instance. Then the loss function is calculated as:

$$Loss_{\mathrm{SEDB}} = \sum_{c} \sum_{t} cross\_entropy(y_{ct}, \hat{\mathbf{P}}(y_{ct}|\mathbf{x}_t)) \tag{7}$$

where c denotes event category, t denotes frame number, $\hat{\mathbf{P}}(y_{ct}|\mathbf{x}_t)$ is the instance-level probability output by the SEDB and y_{ct} is the instance-level ground truth. While using the strong labels of the synthetic data to train the SEDB, we also use the weak labels of synthetic data to train other branches of the MBL-GL model. In our method, we only apply the SEDB in PS-model because the PS-model is mainly used to detect the sound events.

3.4. Combination of sound separation and SED

To incorporate sound separation into SED, we train SED models by the separated data output from the baseline system of sound separation. We use the MBL-GL model with instance-level GAP auxiliary branch (no SEDB) as the SS-SED model. Then, we fuse the SED results of models trained by real data and separated data to get the final SS-SED-Ensemble system result (For SS-SED model, the data should be separated and then be used for the SS-SED model at the inference stage).

$$\begin{aligned}\hat{\mathbf{P}}^{\mathrm{SS-SED-Ensemble}}(y_{ct}|\mathbf{x}_t) = \sum_{i} w_i \hat{\mathbf{P}}_i^{\mathrm{SED}}(y_{ct}|\mathbf{x}_t) + \\ \sum_{j} w_j \hat{\mathbf{P}}_j^{\mathrm{SS-SED}}(y_{ct}|\mathbf{x}_t^{\mathrm{SS}})\end{aligned} \tag{8}$$

where $\sum_i w_i + \sum_j w_j = 1$ and $\mathbf{x}_t^{\mathbf{SS}}$ is the feature of separated data.

4. EXPERIMENTAL SETUP

4.1. Model architecture

As shown in Figure 2, for the PS-model, the feature encoder consists of 3 CNN blocks, each of which contains a convolutional layer, a batch normalization layer, a max pooling layer and a ReLU activation layer. For the PT-model, the feature encoder consists of 8 CNN blocks with dropout. The main branch of the PS-model and the PT-model uses embedding-level ATP. And the PS-model has an auxiliary branch which uses instance-level GMP or GAP. The SEDB is optional and is added to the PS-model in some systems.

4.2. Data

The training set of our system contains a weakly-labeled set (1578 clips), an unlabeled set (14412 clips), and a strongly labeled synthetic set (2584 clips). The validation set contains 1168 strongly-labeled clips. The public test set contains 692 strong-labeled clips. Data augmentation is also applied in the training process. For all training data, we use time-shifting and frequency-shifting to generate augmented data. For time-shifting, all frames (500) are shifted for exactly 90 steps. For frequency-shifting, all frequencies (64) are shifted for exactly 8 steps. The ratio between original data and augmented data is 8:1.

4.3. Model training

We use the Adam optimizer with learning rate 0.0018 to train the model. The learning rate is reduced by 20% for every 10 epochs. The mini-batch size is set to be 64. For a mini-batch of data, we set the ratio of the weakly-labeled data:synthetic data:unlabeled data to be 3:1:12 (It means that there are 12 weakly-labeled clips, 4 synthetic clips and 48 unlabeled clips in a mini-batch).

We report the event-based marco F1 score [8]. All the experiments are repeated 20 times with random initiation and we report both the average result and the best result of each model.

4.4. System ensemble

For system ensemble, we choose different kinds of systems to construct the ensemble system. We take 3 systems with instance-level GAP as auxiliary branch and 3 systems with instance-level GMP as auxiliary branch to construct the SED-Ensemble system. To make the difference between systems large enough, 2 of the 3 systems with instance-level GMP auxiliary branch are with SEDB. To construct the SS-SED-Ensembel system (SS denotes sound separation), besides the 6 systems in the SED-Ensemble system, we add 3 other systems which are trained by sound separated data and has instance-level GAP auxiliary branch. We take the weighted sum of all the system outputs as the final results. The ensembling function is:

$$\hat{\mathbf{P}}^{\mathrm{ensemble}}(y_{ct}|\mathbf{x}_t) = \sum_{i} w_i \hat{\mathbf{P}}_i^{\mathrm{single-system}}(y_{ct}|\mathbf{x}_t) \tag{9}$$

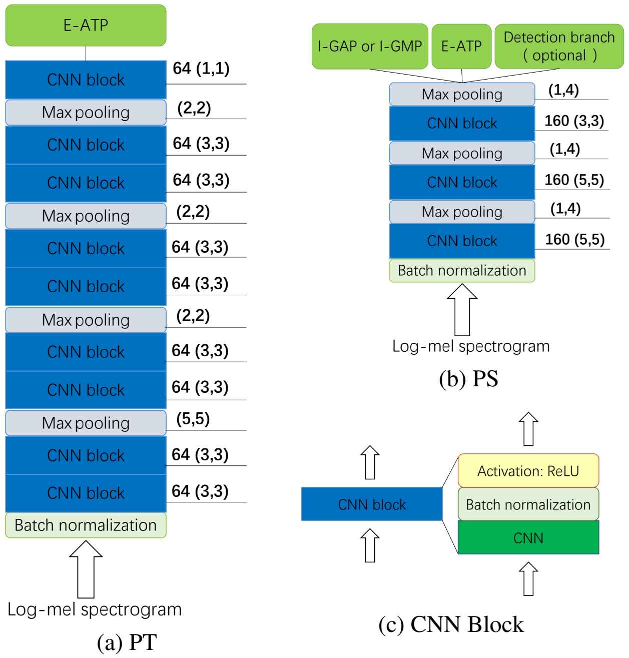
where $\sum_i w_i = 1$. The default value of w_i may be $1/number\_of\_systems$ and in our work, the values are tuned based on the performance on the validation set. Multiple systems with the same model are just trained with different random initializations.

4.5. The competition systems

We participated in 2 subtasks which are SED without SS and SED with SS. We submitted 4 systems for each subtask and the best system for subtask 1 achieves an event-based F1 score of 44.6% and the best system for subtask 2 achieves an event-based F1 score of 44.7%. For our best system of subtask 1, we use 6 systems to make the ensemble system. For our best system of subtask 2, besides the 6 systems in subtask 1, we use 3 systems trained by sound separated data to make the ensemble system.

5. EXPERIMENT RESULTS

Experimental results are shown in Table 1, Table 2 and Table 3. Table 1 shows the results of individual systems which are used for system ensembling, and Table 2 and Table 3 show the average and the best results of each kind of system and the results of ensemble systems. For all the experiments, we do not change the PT model and only change the PS models. In the tables, E-* denotes the embedding-level approach and I-* denotes the instance-level approach. SS-* denotes the system uses the sound separated data. For the baseline system E-ATP, we use the MBL-GL model structure. The only difference between the baseline system and other two kinds of system (E-ATP + I-GAP, E-ATP + I-GMP) is that the baseline system dose not have any auxiliary branch. As shown in Table 2 and Table 3, we find that adding auxiliary branch such as



(a) PT

(b) PS

(c) CNN Block

Figure 2: The model architectures

Table 1: The event-based F1 of individual systems

Model	validation	public test
E-ATP + I-GAP-1	0.447	0.463
E-ATP + I-GAP-2	0.448	0.466
E-ATP + I-GAP-3	0.451	0.461
E-ATP + I-GMP-1	0.450	0.466
E-ATP + I-GMP-2 (with SEDB)	0.448	0.477
E-ATP + I-GMP-3 (with SEDB)	0.454	0.474
SS - E-ATP + I-GAP-1	0.378	0.404
SS - E-ATP + I-GAP-2	0.381	0.390
SS - E-ATP + I-GAP-3	0.378	0.394
E-ATP + I-GMP-4 (not submitted)	0.451	0.473
E-ATP + I-GMP-5 (not submitted)	0.449	0.474
E-ATP + I-GAP-4 (not submitted)	0.414	0.441
E-ATP + I-GAP-5 (not submitted)	0.417	0.439
E-ATP + I-GAP-6 (not submitted)	0.429	0.439

I-GMP or I-GAP can have a beneficial effect. For the ensemble system, we use 3 E-ATP + I-GMP and 3 E-ATP + I-GAP systems to construct it. Besides, to make the difference between models larger, 2 of the 3 E-ATP + I-GMP models use SEDB. The ensemble system achieves an F1 score of 0.497 on the public test set and 0.467 on the validation set. Compared to only using systems without the SEDB, using some systems with the SEDB has potential to improve the performance of the ensemble system: We use the 3 E-ATP + I-GAP and 3 E-ATP + I-GMP systems (2 of the 3 systems which use SEDB are replaced by E-ATP + I-GMP-4 and E-ATP + I-GMP-5) without SEDB to construct ensemble system which is named SED-Ensemble_6_systems. It achieves F1 scores of 0.495 on public test set and 0.463 on the validation set, which are not as good as the ensemble system using SEDB, i. e., SED-Ensemble (submitted).

For the SS-SED-Ensemble system, besides the 6 models used in SED-Ensemble, 3 E-ATP + I-GAP models which are trained by separated data are used and the SS-SED ensemble system achieves F1 scores of 0.495 on the public test set and 0.472 on the validation set. We use 3 other E-ATP+I-GAP systems (E-ATP + I-GAP-4, E-ATP + I-GAP-5, E-ATP + I-GAP-6) trained by real data to replace the SS-SED systems in the SS-SED-Ensemble system. The ensemble system which is named SED-Ensemble_9_systems achieves F1 scores of 0.485 on the public test set and 0.463 on validation set, which are lower than the SS-SED-Ensemble system. Although the performances of the 3 E-ATP + I-GAP systems trained by real data are better than the SS-E-ATP + I-GAP systems, they can not improve the system performance while the SS-E-ATP + I-GAP can improve the performance. It proves that adding SS-SED systems to construct the ensemble system can achieve a better performance since the SS-SED systems may have some feature that SED systems do not have. If the performance of the SS-SED system can be further improved, it is expected to further improve the performance of the ensemble system.

Table 2: The event-based F1 scores on the validation set

Model	Average F1	Best F1
E-ATP	0.421 ± 0.0115	0.444
E-ATP + I-GMP	0.430 ± 0.0088	0.445
E-ATP + I-GAP	0.431 ± 0.0156	0.451
SED-Ensemble (submitted)	-	0.467
SED-Ensemble_6_systems	-	0.463
SS-SED-Ensemble (submitted)	-	0.472
SED-Ensemble_9_systems	-	0.463

Table 3: The event-based F1 scores on the public test set

Model	Average F1	Best F1
E-ATP	0.449 ± 0.0124	0.47
E-ATP + I-GMP	0.458 ± 0.0125	0.478
E-ATP + I-GAP	0.450 ± 0.0130	0.470
SED-Ensemble (submitted)	-	0.497
SED-Ensemble_6_systems	-	0.495
SS-SED-Ensemble (submitted)	-	0.495
SED-Ensemble_9_systems	-	0.485

6. CONCLUSIONS

This paper presents the details of our systems for DCASE 2020 task 4. The systems are based on the first-place system of DCASE 2019 task 4, which adopts the multiple instance learning framework with embedding-level attention pooling and a semi-supervised learning approach called guided learning. The multi-branch learning approach is then incorporated into the system to further improve the performance. Multiple branches with different pooling strategies and different pooling modules are used and share the same feature encoder. To better exploit the synthetic data, inspired by multi-task learning, a sound event detection branch is also added. Therefore, multiple branches pursuing different purposes and focusing on different characteristics of the data can help the feature encoder model the feature space better and avoid over-fitting. The sound separation method is also used and we find that combining the sound separation method to make the ensemble system has great potential to improve the system performance.

7. REFERENCES

[1] http://dcase.community/challenge2020/task-sound-event-detection-and-separation-in-domestic-environments.

[2] http://http://dcase.community/challenge2019/task-sound-event-detection-in-domestic-environments.

[3] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning convolution system for dcase 2019 task 4," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019, pp. 134–138.

[4] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized decision surface and disentangled feature for weakly-supervised polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1466–1478, 2020.

[5] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 626–630.

[6] Y. Huang, X. Wang, L. Lin, H. Liu, and Y. Qian, "Multi-branch learning for weakly-labeled sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 641–645.

[7] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. Huang, "Multi-task learning for acoustic event detection using event and frame position information," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 569–578, 2020.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.