JOINT TRAINING OF GUIDED LEARNING AND MEAN TEACHER MODELS FOR SOUND EVENT DETECTION

Hao Yen[1,2], *Pin-Jui Ku*[1,2], *Ming-Chi Yen*[1], *Hung-Shin Lee*[1,2], *Hsin-Min Wang*[1]

[1] Institute of Information Science, Academia Sinica, Taiwan
[2] Department of Electrical Engineering, National Taiwan University, Taiwan
{b05901090, b05901107}@ntu.edu.tw

ABSTRACT

In this paper, we present our system of sound event detection and separation in domestic environments for DCASE 2020. The task aims to determine which sound events appear in a clip and the detailed temporal ranges they occupy. The system is trained by using weakly-labeled and unlabeled real data and synthetic data with strongly annotated labels. Our proposed model structure includes a feature-level front-end based on convolution neural networks (CNN), followed by both embedding-level and instance-level back-end attention modules. In order to make full use of the large amount of unlabeled data, we jointly adopt the Guided Learning and Mean Teacher approaches to carry out weakly-supervised learning and semi-supervised learning. In addition, a set of adaptive median windows for individual sound events is used to smooth the frame-level predictions in post-processing. In the public evaluation set of DCASE 2019, the best event-based F_1-score achieved by our system is 48.50%, which is a relative improvement of 27.16% over the official baseline (38.14%). In addition, in the development set of DCASE 2020, our best system also achieves a relative improvement of 32.91% over the baseline (45.68% vs. 34.37%).

Index Terms— Guided learning, Mean teacher, Semi-supervised learning.

1. INTRODUCTION

DCASE 2020 Task 4 is a follow-up to DCASE 2019 Task 4, which aims to develop a sound event detection (SED) system that can predict not only the presence of events, but also the onset and offset positions of each event. The challenge provides three types of data, namely, weakly-labeled data (without timestamps), unlabeled data, and synthetic data with strong annotations (with timestamps). Each 10-second audio clip contains one or more (or none) of 10 sound events, including alarm bell ringing, blender, cat, dishes, dog, electric shaver, frying, running water, speech, and vacuum cleaner. The training set contains much more unlabeled data than labeled data. In addition, the number of training clips for each label is unbalanced.

Traditional approaches of SED often adopt deep neural networks such as CNN [1, 2, 3], recurrent neural network (RNN) [4], or convolutional recurrent neural network (CRNN) [5], and usually require a lot of strongly annotated real data, making them unsuitable for this task. Therefore, the main focus of the task is to effectively exploit unlabeled training data, and to mitigate the impact of label preference during training to achieve better test performance.

To deal with the aforementioned problems, previous methods tend to adopt weakly-supervised or semi-supervised learning techniques [6, 7]. Recently, the teacher-student structure is commonly used in the task. In DCASE 2019, Guided Learning [8] introduced us a brand new weakly-labeled semi-supervised learning algorithm. It utilized a more professional teacher model designed for audio tagging to guide the student model to learn from unlabeled data for boundary detection. However, the system did not involve learning from data with strongly annotated timestamp information.

Meanwhile, Mean Teacher [9] is another state-of-the-art approach for semi-supervised learning in the task. In DCASE 2019 Task 4, the Mean Teacher based system was the runner-up [10], while in DCASE 2020 Task 4, its improved version became the official baseline system. With the consistency loss, Mean Teacher can learn not only from weakly and strongly annotated data, but also from unlabeled data. Using unlabeled real data also prevent the system from overfitting the strongly annotated synthetic data. However, as we observed, using Mean Teacher to achieve better performance usually requires a very robust representation, which the current Mean Teacher-based system cannot provide.

In this paper, we present a unified approach to sound event detection, which combines the best methods of the past. In the first training step, Guided Learning learns a well-trained CNN front-end, which can convert the input log Mel-spectrogram into an informative high-level representation. In the second training step, Mean Teacher makes full use of strongly annotated information to train the recurrent neural network (RNN) and frame-based scorer. By making full use of weakly annotated real data, strongly annotated synthetic data and unlabeled real data, our model achieves competitive results in both audio tagging and boundary detection. In addition, we also use a set of event-dependent median windows to further improve boundary detection by smoothing the frame-level predictions of each event in post-processing.

The remainder of this paper is organized as follows. Section 2 introduces our proposed method for sound event detection, including the model structure, learning process, and adaptive issues. More detailed information about Mean Teacher and Guided Learning is also provided. Section 3 presents the experimental setup and results. Finally, we conclude this paper in Section 4.

2. PROPOSED METHOD

Our system is developed based on the official baseline system, which is based on Mean Teacher [9, 10] and is an improved version of the second best submission system of DCASE 2019 Task 4. Figure 1 shows the overall flowchart of our system. We train our model with two training steps, namely Guided Learning [8, 11, 12] (Step 1) and Mean Teacher (Step 2). The details of the two training steps and model structures will be explained in the following sections.

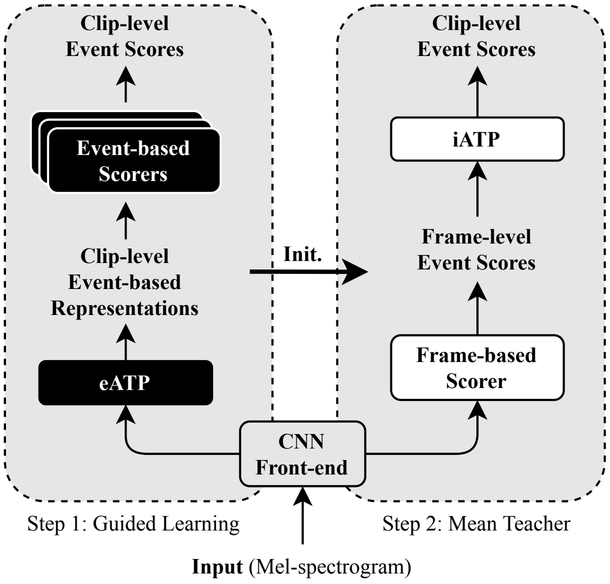


Figure 1: *Flowchart of our system. The CNN front-end is pre-trained by the Guided Learning algorithm in Step 1. In Step 2, Mean Teacher learning is used to fine-tune the pre-trained CNN front-end and train the frame-based scorer and iATP simultaneously.*

2.1. Model structures

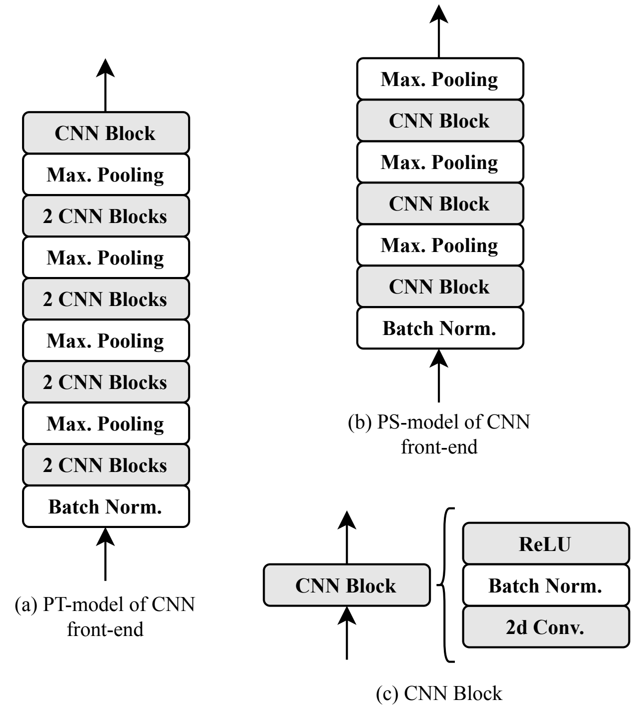
As shown in Figure 1, the model consists of a CNN front-end, which aims to generate robust high-level representations, followed by several scoring and attention pooling modules. The model is trained in two training steps. In Step 1, an embedding-level attention pooling module (eATP) converts the high-level representations into a clip-level representation, which is then used by individual event-based scoring modules to predict the clip-level event scores. In Step 2, a frame-based scorer is connected after the CNN front-end to generate frame-level event scores, and then an instance-level attention pooling module (iATP) produces the final clip-level event scores. The detailed model structures are shown in Figures 2 and 3. We utilize different training algorithms in the two steps. For Step 1, we follow the Guided Learning framework in [11] and use a more professional teacher model to carry out weakly-supervised learning. As for Step 2, we apply the Mean Teacher [9] method for semi-supervised learning.

2.1.1. CNN front-ends

The CNN front-end adopts the same structure as in [8], as shown in Figure 2(b). It consists of a batch normalization layer [13] and three CNN blocks (cf. Figure 2(c)). Each CNN block has a single 2-dimensional CNN layer, a batch normalization layer, and an ReLU activation layer. A Max-pooling layer comes after each CNN block. According to [8], the CNN-based structure can convert the input log Mel-spectrogram into a robust high-level representation, which is then passed to the pooling module. In our implementation, we pre-train the CNN front-end using Guided Learning (cf. Step 1), and then integrate it into the training process of Step 2.

2.1.2. Pooling modules

In [12, 14, 15], the effect of pooling on the SED task is highlighted. The embedding-level pooling module directly aggregates the high-level feature representations into an event-based representation (cf. eATP in Step 1 in Figure 1). The embedding-level pooling approach is superior to instance-level pooling in general, so it is adopted in Guided Learning. Since the strongly annotated data are not used in the training process, it relies heavily on the CNN front-end prior to eATP (cf. Figure 2(b)) to learn frame-level information, thereby resulting in a stronger front-end. On the other hand, for the instance-level approach, the high-level feature representations are passed to the classifier to generate frame-level event scores. Then, the pooling module aggregates frame-level scores into a clip-level event score. The instance-level pooling approach can take advantage of strongly annotated timestamp information by calculating the loss between the frame-level event scores and the ground truth, but usually requires a more powerful front-end to generate accurate scores.



(a) PT-model of CNN front-end

(b) PS-model of CNN front-end

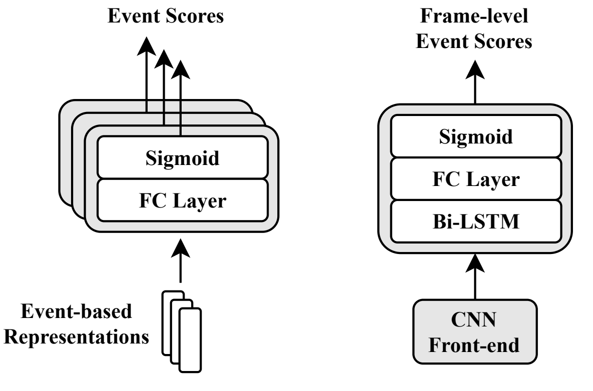
(c) CNN Block

Figure 2: *The CNN front-ends used in Guided Learning (Step 1), where the PT-model and the PS-model are two key modules. Afterwards, the PS-model is taken as the initialization of the CNN front-end in Mean Teacher learning (Step 2).*

We argue that by training the two pooling modules in turn, the overall performance on both sides can be improved. The specific procedures are as follows. First, we use Guided Learning with embedding-level pooling to obtain more robust and abstract representations for instance-level pooling to generate better frame-level event scores. Next, we utilize the strongly labeled information through instance-level pooling to further fine-tune the front-end. As shown in Figures 2 and 3, we adopt the same eATP structure and event-based scorers in [12]. The same RNN structure and pooling module as in the baseline system are used for the frame-level event scorers and iATP in our model.

2.2. Learning processes

In this section, we explain the training process of our model. First, we introduce two learning techniques, i.e., Guided Learning and



(a) Event-based Scorers (b) Frame-based Scorer

Figure 3: *Model structures of the scorers. Each event-based scorer of Step 1 in (a) consists of a fully-connected (FC) layer and a sigmoid-based activation function (Sigmoid). The frame-based scorer of Step 2 in (b) consists of a Bi-LSTM, an FC Layer, and Sigmoid.*

Mean Teacher. Then, we describe how to integrate these two techniques into our model learning process.

2.2.1. Guided Learning

As proposed in [8], Guided Learning consists of a teacher model (PT-model) and a student model (PS-model), which are shown in Figure 2(a) and (b). The PT-model has a deeper CNN front-end structure and a larger receptive field than the PS-model. Therefore, we can foresee that the PT-model will yield better audio tagging performance.

Nevertheless, the larger receptive field is accompanied by greater time compression in the PT-model, thus reducing the ability of the model to see finer information hidden in the time dimension. Therefore, the PS-model is designed not to perform time compression in order to obtain better performance in frame-level prediction.

Due to their different abilities in clip-level and frame-level predictions, we can make use of unlabeled data by making the PS-model learn from the pseudo labels [16] generated by the PT-model.

2.2.2. Mean Teacher

As stated in [9], the main purpose of the Mean Teacher approach is to average the model weights after each training step, e.g., to use exponential moving average to produce a more accurate model instead of directly using the latest model weights. We call the average model as the Mean Teacher model (MT-model) and the latest model as the Mean Student model (MS-model). In each training step, we calculate two kinds of losses: the classification loss and the consistency loss.

For the classification loss, we compute the binary cross entropy from the predictions of the MS-model for the labeled data. As for the consistency loss, it can be obtained by comparing the clip-level and frame-level predictions given by the MS-model and the MT-model for all the labeled and unlabeled data. In other words, we want the MS-model and the MT-model to output similar predictions for the same clip. The two losses are summed to update the MS-model. Then, the MT-model is updated by the new average weights.

Table 1: *Median window sizes (S_{win}) with respect to sound events.*

Event	S_{win}	Event	S_{win}
Alarm bell ringing	18	Electric shaver	161
Blender	52	Frying	196
Cat	29	Running water	80
Dishes	11	Speech	18
Dog	15	Vacuum cleaner	177

2.2.3. The GL-MT learning algorithm

The aforementioned CNN front-end is first pre-trained using Guided Learning (Step 1 in Figure 1). After normalizing the input log Mel-spectrograms of the real and synthetic training data separately, we follow the process in Sec. 2.2.1 to train the CNN front-end. To guarantee the ability of the PT-model, we use both weakly annotated real data and strongly annotated synthetic data to train the PT-model with a supervised loss with respect to clip-level event scores. Then, the PS-model can be trained by using not only the same supervised loss but also an unsupervised loss, where the tags predicted by the PT-model for unlabeled data are considered as the ground truth labels. Note that we do not adopt the feature disentanglement method proposed in [12]. That is, all categories share the same feature space of the extracted high-level representation. The last layer of the CNN front-end from the PS-model can be an informative representation, which is applied to frame-based scorers in the Mean Teacher model (cf. Step 2).

After the CNN front-end is well-trained to be able to extract robust representations, Mean Teacher is used to simultaneously train the Mean Teacher model and fine-tune the CNN front-end with the strongly annotated training data. We calculate the classification loss based on both clip-level and frame-level event scores. The model can also exploit unlabeled data through calculating the consistency loss between the predicted scores of the MS-model and MT-model.

2.3. Event detection

In our system, we take the mean of the clip-level event scores from the Guided Learning model and the Mean Teacher model as the final clip-level event scores, and use a threshold of 0.5 for 0/1 prediction.

The frame-level 0/1 prediction at time t is determined by

$$F(\mathbf{x}, t) = p(x_t) \cdot C(\mathbf{x}), \tag{1}$$

where $p(x_t)$ denotes the frame-level event scores in Figure 3(b), and $C(\mathbf{x})$ represents the above mean clip-level scores. For an event, if the corresponding $F(\mathbf{x}, t)$ is larger than the threshold, then the output of 0/1 prediction will be 1. This threshold is also set to 0.5.

2.4. Adaptive median windows

A median filter can be used to post-process the frame-level output. Once the frame-level event scores are generated by our system, they will be smoothed by event-dependent median windows before being converted into 0/1 prediction with the threshold of 0.5. We will then smooth the resulting 0/1 prediction sequence with the same set of median windows. In [12], the importance of median filtering is underlined. Instead of using a fixed-size window for every class as in the baseline system, we design a specific median window for each event so that each class has its own unique window. The idea is to take into account the duration of each category in the dataset, and to obtain more accurate boundaries by using median windows

Table 2: *F_1-scores with respect to various models with a fixed median window size of 33.*

	Event-based		**Segment-based**	
Model	**Dev**	**Eval**	**Dev**	**Eval**
Baseline	34.37	38.14	69.07	71.68
GL-ps	37.78	37.26	70.01	72.44
GL-MT-ps	38.66	39.90	67.16	68.91
GL-MT-ms	40.96	42.20	70.83	73.35
GL-MT-ema	**41.12**	**44.40**	**71.06**	**74.70**

of appropriate length. To determine the size of the event-dependent median window, we analyze the average duration of each event category in the validation and synthetic sets. We follow [8] and calculate the window size S_{win} as:

$$S_{win} = D_{avg} \times \beta, \quad (2)$$

where $\beta = 1/3$, and D_{avg} denotes the average duration of a class in the dataset. Then, we make small adjustments to the window sizes given by Eq. 2 based on the validation results. Table 1 shows the final window size of each event.

3. EXPERIMENTS AND RESULTS

3.1. Dataset

We utilize the dataset provided by DCASE 2020 as our training dataset, which consists of 3 subsets: weakly-annotated data (1,578 clips), unlabeled data (14,412 clips), and strongly-annotated data (2,584 clips). The weakly-labeled and unlabeled data are real data with a sampling rate of 44,100Hz, while the strongly-annotated data are synthetic data with a sampling rate of 16,000Hz.

In DCASE 2020 Task 4, the event-based F_1-score (macro-average) [17] is used to evaluate the performance. We take the 1,168 clips from the validation set provided by DCASE 2020 as our development set and the 692 clips from the public evaluation set provided in DCASE 2019 as our evaluation set. The validation and evaluation data are real data with a sampling rate of 44,100Hz. All the data at 44,100Hz are downsampled to 16,000Hz in this work. We report both event-based and segment-based (1s) detection results.

3.2. Training

In Step 1, we utilize a mini-batch of 32 10-second clips and the Adam optimizer [18] with an initial learning rate of 0.0018 to train our model for 100 epochs. The learning rate is reduced by 20% every 10 epochs. The same optimizer with a lower initial learning rate of 0.001 is then adopted for Step 2 training. We evaluate the event-based F_1-score after each epoch on the development set, and store the best model accordingly.

3.3. Results

The original Guided Learning-based system is named GL-ps, which uses the PS-model as the detector. Our approaches of combining Guided Learning and Mean Teacher are named GL-MT-ms and GL-MT-ps, which use the MS-model and the PS-model as the detector, respectively. In addition, we also use the exponential moving average (EMA) model from Mean Teacher in Step 2 as the detector. This model is named GL-MT-ema. We compare three GL-MT-based models with the baseline model provided by DCASE 2020 and the GL-ps model.

Table 3: *F_1-scores with respect to various models with adaptive median window sizes.*

	Event-based		**Segment-based**	
Model	**Dev**	**Eval**	**Dev**	**Eval**
GL-ps	45.05	42.53	70.81	72.34
GL-MT-ps	45.42	45.41	69.04	70.93
GL-MT-ms	**45.68**	47.47	**71.96**	74.63
GL-MT-ema	45.65	**48.50**	71.87	**75.83**

Table 2 shows the F_1-scores of different models with a fixed-size median window for post-processing. For the baseline system, the size is 28 (it is 7, but is equivalent to 28 considering the time compression factor). For the other models, the size is empirically set to 33 according to the validation results. From Table 2, we can see that GL-MT-ms and GL-MT-ema outperform the baseline and GL-ps, which are based on Mean Teacher and Guided Learning, respectively. However, GL-MT-ps is not always better than the baseline and GL-ps. GL-MT-ms consists of the same RNN-based frame-based scorer and instance-level pooling module as the official baseline system and a more robust CNN front-end. The results support our argument that a better CNN front-end can produce more useful high-level representations for the frame-based scorer to generate more accurate frame-level event scores. Overall, the experimental results confirm the effectiveness of combining Guided Learning and Mean Teacher for sound event detection.

Next, we evaluate effectiveness of the adaptive median windows for post-processing. The results are shown in Table 3. Comparing Table 3 with Table 2, we can see that all the models with adaptive median windows are superior to their counterparts with a fixed-size median window. The results confirm the effectiveness of the adaptive media windows for post processing. GL-MT-ema achieves the best performance, with relative improvements of 27.16% (48.50% vs. 38.14%) and 5.79% (75.83% vs. 71.68%) in terms of event-based and segment-based F_1-scores over the baseline on the evaluation set.

4. CONCLUSIONS

This paper presents our submission systems for DCASE 2020 Task 4. We utilize a CNN-based front-end with different pooling modules and scorers, including embedding-level attention pooling with event-based scorers and frame-based scorers with instance-level attention pooling. We combine Guided Learning and Mean Teacher methods to carry out weakly-supervised and semi-supervised learning. We perform the two training steps in sequence. The first training step pre-trains a robust CNN front-end to provide more informative high-level representations for the second training step to train the back-end detector. It is confirmed that joint Guided Learning and Mean Teacher training is superior to the respective single training method. In addition, we adopt adaptive median windows for post-processing. The experimental results show that adaptive median windows can produce more accurate event boundaries than fixed-size median windows.

5. REFERENCES

[1] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time-frequency segmentation from weakly labelled data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 777–787, 2019.

[2] T.-w. Su, J.-y. Liu, and Y.-h. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *Proc. ICASSP*, 2017.

[3] A. Kumar, M. Khadkevich, and C. Fugen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *Proc. ICASSP*, 2018.

[4] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. ICASSP*, 2016.

[5] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 1291 – 1303, 2017.

[6] T. Hua, F. Chen, L. Zhao, C. T. Lu, and N. Ramakrishnan, "STED: Semi-supervised targeted-interest event detection in twitter," in *Proc. SIGKDD*, 2013.

[7] B. Shi, M. Sun, C. C. Kao, V. Rozgic, S. Matsoukas, and C. Wang, "Semi-supervised acoustic event detection based on tri-training," in *Proc. ICASSP*, 2019.

[8] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning convolution system for DCASE 2019 Task 4," in *Proc. DCASE*, 2019.

[9] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. NIPS*, 2017.

[10] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 Task 4," in *Proc. DCASE*, 2019.

[11] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," in *Proc. ICASSP*, 2020.

[12] ——, "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.

[13] S. Ioffe and C. Szegedy, "Batch normalization:accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[14] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. ICML*, vol. 5, 2018.

[15] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[16] Dong-Hyun Lee, "Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. ICML*, 2013.

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

[18] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015.