WEAKLY LABELED SOUND EVENT DETECTION USING TRI-TRAINING AND ADVERSARIAL LEARNING

Hyoungwoo Park, Sungrack Yun, Jungyun Eum, Janghoon Cho, Kyuwoong Hwang

Qualcomm AI Research*, Qualcomm Korea YH
{c_hyoupa, sungrack, c_jeum, janghoon, kyuwoong}@qti.qualcomm.com

ABSTRACT

This paper considers a semi-supervised learning framework for weakly labeled polyphonic sound event detection problems for the DCASE 2019 challenge's task4 by combining both the tri-training and adversarial learning. The goal of the task4 is to detect onsets and offsets of multiple sound events in a single audio clip. The entire dataset consists of the synthetic data with a strong label (sound event labels with boundaries) and real data with weakly labeled (sound event labels) and unlabeled dataset. Given this dataset, we apply the tri-training where two different classifiers are used to obtain pseudo labels on the weakly labeled and unlabeled dataset, and the final classifier is trained using the strongly labeled dataset and weakly/unlabeled dataset with pseudo labels. Also, we apply the adversarial learning to reduce the domain gap between the real and synthetic dataset. We evaluated our learning framework using the validation set of the task4 dataset, and in the experiments, our learning framework shows a considerable performance improvement over the baseline model.

Index Terms— Sound event detection (SED), Tri-training, Pseudo labeling, Adversarial learning, Semi-supervised learning, Weakly supervised learning

1. INTRODUCTION

The polyphonic sound event detection (SED) has been attracting growing attention in the field of acoustic signal processing [1–8]. The SED aims to detect multiple sound events happened simultaneously as well as the time frame in a sequence of audio events. The applications of the SED include audio event classification [9–11], media retrieval [12, 13] and automatic surveillance [11] in living environments such as Google Nest Cam [14] which analyzes the audio stream to detect conspicuous sounds such as window breaking and dog barking among various sounds that could occur in daily environments.

Several researches [2, 7, 8, 10, 15–18] have been previously proposed . In [4], spectral domain features are used to characterize the audio events, and deep neural networks (DNN) [10] is used to learn a mapping between the features and sound events. In [18], multiple instance learning was exploited to predict the labels of new, unseen instances which rely on an ensemble of instances, rather than individual instances. In [5], convolutional recurrent neural networks (CRNN) was introduced which is a combined network of convolutional neural networks (CNN) [15, 16] and recurrent neural networks (RNN) [8] to get the benefits of both CNN and RNN. In [19], Mean Teacher method was adopted where the teacher model is an average of consecutive student models to overcome the limitations of temporal ensembling for semi-supervised learning.

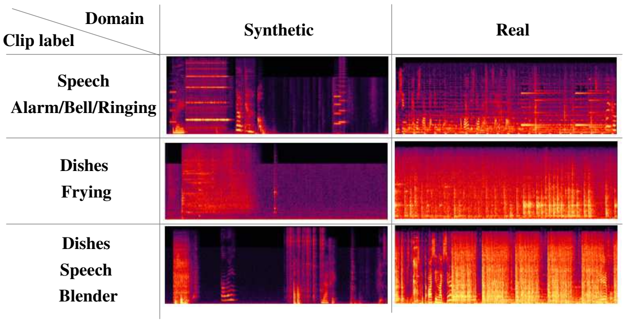

Figure 1: The spectrograms of synthetic and real dataset samples which have the same clip labels

In contrast to the task 4 of the last year's challenge [20], a synthetic dataset with strong annotation is additionally provided in DCASE 2019 challenge's task 4. Strong annotation includes onset, offset and class label of the sound events. Thus, how to utilize strongly-labeled synthetic data and mutual complement between real dataset and synthetic dataset is a challenging problem in weakly labeled SED problem in DCASE 2019 challenge's task 4. Previous methods have not focused on complement of strongly labeled synthetic dataeset. As shown in Figure 1, the log-mel spectrogram of samples which have the same clip label from synthetic data and real data seem too much different. For this reason, we assume that domain gap between synthetic data and real data exists and it causes degradation of performance on test samples.

This paper presents a sound event detection combining adversarial learning and tri-training. Adversarial learning helps to reduce the gap between synthetic and real data by learning domain-invariant feature while tri-training method [21] which is one of the semi-superived learning methods learns discriminative representations by pseudo labeling one the weakly labeled or unlabeled samples. Pseudo labels are obtained by agreement of output from confident two labelers on unlable data. Inspired by these properties, we present a weakly labeled polyphonic SED by considering both adversarial learning and tri-training.

The proposed learning framework was evaluated using a validation set of the DCASE 2019 challenge's task 4 [22]. In the evaluation results, combined adversarial training and tri-training shows a considerable performance improvement over the baseline model.

[*] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

https://doi.org/10.33682/ntr9-6764

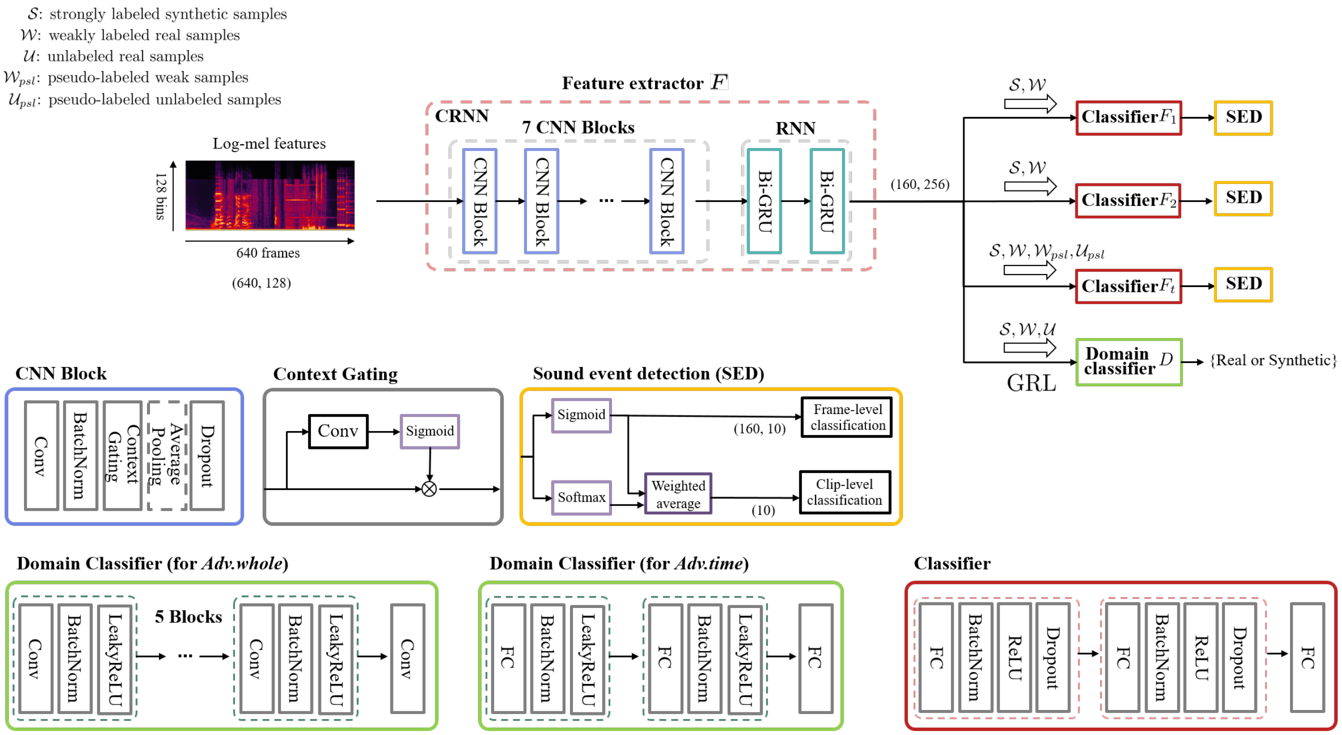
$\mathcal{S}$: strongly labeled synthetic samples
$\mathcal{W}$: weakly labeled real samples
$\mathcal{U}$: unlabeled real samples
$\mathcal{W}_{psl}$: pseudo-labeled weak samples
$\mathcal{U}_{psl}$: pseudo-labeled unlabeled samples
Log-mel features
128 bins
640 frames
(640, 128)
Feature extractor F
CRNN
7 CNN Blocks
CNN Block
CNN Block
CNN Block
RNN
Bi-GRU
Bi-GRU
(160, 256)
$\mathcal{S}, \mathcal{W}$
Classifier F_1
SED
$\mathcal{S}, \mathcal{W}$
Classifier F_2
SED
$\mathcal{S}, \mathcal{W}, \mathcal{W}_{psl}, \mathcal{U}_{psl}$
Classifier F_t
SED
$\mathcal{S}, \mathcal{W}, \mathcal{U}$
Domain classifier D
{Real or Synthetic}
GRL
CNN Block
Conv
BatchNorm
Context Gating
Average Pooling
Dropout
Context Gating
Conv
Sigmoid
Sound event detection (SED)
Sigmoid
Softmax
Weighted average
Frame-level classification
(160, 10)
Clip-level classification
(10)
Domain Classifier (for Adv.whole)
Conv
BatchNorm
LeakyReLU
5 Blocks
Conv
BatchNorm
LeakyReLU
Conv
Domain Classifier (for Adv.time)
FC
BatchNorm
LeakyReLU
FC
BatchNorm
LeakyReLU
FC
Classifier
FC
BatchNorm
ReLU
Dropout
FC
BatchNorm
ReLU
Dropout
FC

Figure 2: The proposed learning framework includes feature extractor F, classifiers(pseudo-labelers) F_1, F_2, final classifier F_t and domain classifier D. The dataset to train each component is shown in the figure (e.g. classifier F_2 is trained using the strongly-labeled synthetic samples $\mathcal{S}$ and weakly-labeled real samples $\mathcal{W}$. The pseudo-labels are obtained by agreement from two different classifiers F_1, F_2 and used in training the final classifier F_t. The domain classifier D, connected to F via a GRL, classifies the input feature into real or synthetic. With the GRL from D to F, the feature distributions between synthetic and real domain become similar, and thus we can obtain the domain-invariant features.

2. PROBLEM STATEMENT AND NOTATIONS

For SED, we denote a sound clip by $x \in \mathcal{X}$ and corresponding $y \in \mathcal{Y}$. The SED systems are expected to produce strongly labeled output y^s (i.e. sound class label with start time and end time) from input x. However, for weakly labeled SED with semi-supervised setting, dataset consists of strongly labeled data $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^m$, weakly labeled data $\mathcal{W} = \{(x_j^w, y_j^w)\}_{j=1}^n$ and unlabeled data $\mathcal{U} = \{x_k^u\}_{k=1}^l$. The weakly labeled data does not provide a temporal range of events but sound class labels detected in a clip. We focus on the usage of weakly labeled or unlabeled data and reducing domain gap between synthetic and real data. Thus, we combine the adversarial learning based on gradient reversal layer (GRL) [23] for reducing the domain gap and tri-training method for pseudo-labeling weakly labeled or unlabeled data such that the networks are learned to output discriminative representations on a real dataset.

3. PROPOSED METHOD

Our proposed method is based on *CRNN* [19] model, which showed the first place of the task 4 in the last year's challenge by combining with Mean Teacher algorithm [24]. The whole architecture is shown in Figure 2. A feature extractor F, which cosists of seven CNN blocks and two bi-directional gated recurrent units (Bi-GRU) [25], outputs shared features from log-mel features used as input for four networks. Two labelers F_1, F_2 and classifier F_t predict multiple classes for each time frame and class events for a clip from features extracted by F. Let L_y be the classification loss with frame-level classification loss L_{frame} and clip-level classification loss L_{clip} for multi-label prediction.

$$L_y = L_{frame} + L_{clip} \tag{1}$$

For training with frame-level classification loss and clip-level classfication loss, binary cross entropy (BCE) loss is used with the sigmoid output:

$$L_{clip} = \sum_{i=1}^{N} \sum_{k=1}^{K} [y_{i,k} \log \hat{y}_{i,k} + (1 - y_{i,k}) \log (1 - \hat{y}_{i,k})] \tag{2}$$

$$L_{frame} = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} [y_{i,t,k} \log \hat{y}_{i,t,k} + (1 - y_{i,t,k}) \log (1 - \hat{y}_{i,t,k})] \tag{3}$$

where $y_{i,k}, y_{i,t,k}^i \in [0, 1]$ are the label of sound class k of clip i and the label sound class k at time frame t of clip i, respectively. Also, $\hat{y}_{i,k}$ is the predicted probability of sound class k of clip i, and $\hat{y}_{i,t,k}$ is $\hat{y}_{i,k}$ at time frame t. A domain classifier D classifies features from F into real or synthetic. Based on this architecture, the proposed adversarial learning with tri-training framework for SED will be explained in the next section.

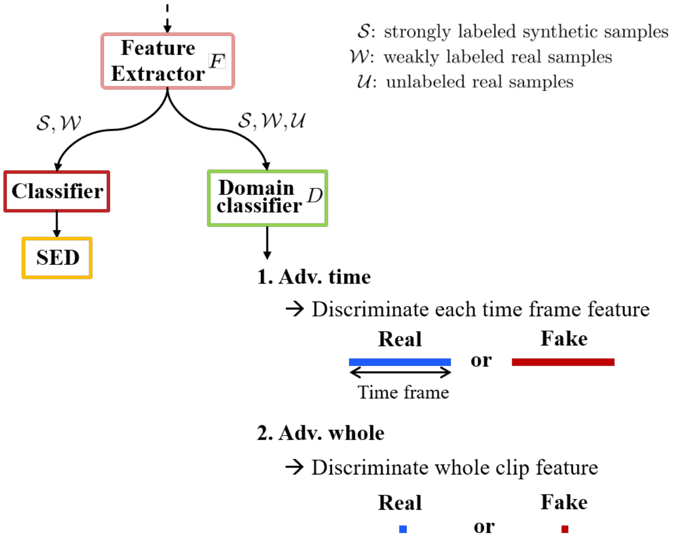
Feature Extractor F
$\mathcal{S}$: strongly labeled synthetic samples
$\mathcal{W}$: weakly labeled real samples
$\mathcal{U}$: unlabeled real samples
$\mathcal{S},\mathcal{W}$
$\mathcal{S},\mathcal{W},\mathcal{U}$
Classifier
Domain classifier D
SED
1. Adv. time
→ Discriminate each time frame feature
Real
or
Fake
Time frame
2. Adv. whole
→ Discriminate whole clip feature
Real
or
Fake

Figure 3: Two approaches of adversarial learning for sound event detection problems

Algorithm 1: The function *Pseudo-labeling* is the process of assigning pseudo-labeling based on agreement threshold from two labelers. We assign pseudo-labels to weakly labeled or unlabeled samples when both predictions of F_1 and F_2 are confident and agreed to the same prediction.

```
Input: strongly labeled synthetic data $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^m$
weakly labeled real data $\mathcal{W} = \{(x_j^w, y_j^w)\}_{j=1}^n$
unlabeled real data $\mathcal{U} = \{(x_k^u)\}_{k=1}^l$
pseudo-labeled weakly labeled data $\mathcal{W}_{psl} = \emptyset$
pseudo-labeled unlabeled data $\mathcal{U}_{psl} = \emptyset$
for $i$ = 1 to iter do
    Train $F, F_1, F_2, F_t, D$ with mini-batch from labeled
      training set $\mathcal{S}, \mathcal{W}, \mathcal{U}$
end
$\mathcal{W}_{psl}$ = Pseudo-labeling($F, F_1, F_2, \mathcal{W}$)
$\mathcal{U}_{psl}$ = Pseudo-labeling($F, F_1, F_2, \mathcal{U}$)
for $j$ =1 to iter do
    Train $F, F_t, D$ with mini-batch from labeled training
      set $\mathcal{S}, \mathcal{W}$ and pseudo-labeled training set $\mathcal{W}_{psl}, \mathcal{U}_{psl}$
end
```

3.1. Adversarial learning

We denote strongly labeled synthetic dataset by $\mathcal{S}$ and weakly labeled or unlabeled real dataset $\mathcal{W}, \mathcal{U}$ are the different domain (synthetic or real). As shown in Figure 1, since the domain gap between the synthetic and real dataset is quite big, we construct a domain classifier to reduce the gap between two domains by adversarial learning. The domain classifier D classifies input features into real or synthetic. By applying the GRL [23] from D to the feature extractor F, we can obtain the feature representation whose distributions are almost similar in both real and synthetic domain. We consider two approaches to apply the adversarial learning for SED as shown in Figure 3. First, D classifies the whole feature from F into one result: real or synthetic (*Adv.whole*). In this case, GRL makes the features from F domain-invariant. Second, D classifies each time frame of the feature into real or synthetic (*Adv.time*). The second approach is more appropriate than the first one since our architecture predicts multiple sound event classes in each time frame from features extracted from F. We denote $\theta_F, \theta_{F_1}, \theta_{F_2}, \theta_{F_t}$ and θ_D by the parameters of each network, respectively. Also, L_d is the loss for the domain classification. For training with domain classfication loss, BCE loss is used with the sigmoid output:

$$L_d = \sum_{i=1}^{N} [d_{i,t} \log \hat{d}_{i,t} + (1 - d_{i,t}) \log (1 - \hat{d}_{i,t})] \tag{4}$$

where $d_{i,t}$ is the label of real or synthetic at time frame t of clip i, and $\hat{d}_{i,t}$ is predicted probability at time frame t of clip i. Based on GRL, the parameters are updated as follows:

$$\theta_F \leftarrow \theta_F - \mu(\frac{\partial L_y}{\partial \theta_F} - \alpha \frac{\partial L_d}{\partial \theta_F}) \tag{5}$$

$$\theta_{F_1,F_2,F_t} \leftarrow \theta_{F_1,F_2,F_t} - \mu \frac{\partial L_y}{\partial \theta_{F_1,F_2,F_t}} \tag{6}$$

$$\theta_D \leftarrow \theta_D - \mu \frac{\partial L_d}{\partial \theta_D} \tag{7}$$

where μ, α are the learning rate and hyperparameter of GRL, respectively.

3.2. Tri-training

We apply the tri-training method to train a network using the pseudo-labeled weakly labeled samples $\mathcal{W}_{psl}$ and pseudo-labeled unlabeled samples $\mathcal{U}_{psl}$. The entire procedure of tri-training is shown in Algorithm 1. First, we train common feature extractor F, two labeling networks F_1 and F_2 , a final classifier F_t and a domain classifier D with labeled samples $\mathcal{S}, \mathcal{W}$ and unlabeled samples $\mathcal{U}$. Second, pseudo-labeled samples are obtained by F_1 and F_2 trained with labeled samples. When the confidences of both networks' outputs exceed the agreement threshold, the prediction can be considered reliable. We set this threshold to 0.5 in the experiments. Also, we expect each labeler to obtain different classifiers F_1 and F_2 given the same training data, we use the following regularization loss:

$$L = L_y + \lambda \left| \left(\frac{W_{F_1}}{|W_{F_1}|} \right)^\top \left(\frac{W_{F_2}}{|W_{F_2}|} \right) \right| \tag{8}$$

where W_{F_1} and W_{F_2} are weights of first layer of two labelers F_1 and F_2, respectively. We set λ to 1.0 based on the validation set. Then, we use both labeled samples $\mathcal{S}, \mathcal{W}$ and pseudo-labeled samples $\mathcal{W}_{psl}, \mathcal{U}_{psl}$ for training F, F_t, and D. Then, F and F_t will learn from the labeled real dataset.

4. EXPERIMENTS

4.1. Dataset

The DCASE 2019 challenge's task 4 [22] provides the following 3 subsets of the dataset in the training: 1,578 clips of the weakly labeled set, 14,412 clips of the unlabeled in-domain set and 2,045 clips of the synthetic set with strong annotations of events and timestamps. Weakly labeled and unlabeled in-domain sets are from

Audioset [26] which drawn from 2 million YouTube videos. The synthetic set is generated with Scaper [27] to increase the variability of the output for soundscape synthesis and augmentation. These audio clips are 10 second-long and contain one or multiple sound events among 10 different classes (speech, dog, cat, alarm/bell/ringing, dishes, frying, blender, running water, vacuum cleaner and electric shaver/toothbrush) which may partly overlap.

4.2. Experimental setup

The model was developed using PyTorch [28] and all experiments were conducted on an a GeForce GTX TITAN X GPU 12GB RAM. Also, our architecture was trained with a mini-batch size of 64 using Adam optimizer [29] with an initial learning rate of 0.001 and exponential decay rate for the $1st$ and $2nd$ moments of 0.9 and 0.999, respectively. The input audio clips are down-sampled from 44.10 kHz to 22.05 kHz. And, the log-mel spectrogram is extracted from the audio clip with the size of 640×128: 128-bin is used, and 2048-window with 345-hop is used to convert into 640 frames.

4.3. Experimental results

We evaluated our proposed framework using the DCASE 2019 challenge's task4 validation dataset. We could not measure performance on evaluation dataset since the labels of evaluation dataset are not available yet. The macro event-based F1 scores and segment-based F1 scores on validation dataset are shown in Table 1. Segment-based metrics evaluate an active/inactive state for each sound event in a fixed-length interval, while event-based metrics evaluate sound event class detected in the fixed-length interval. The baseline of DCASE 2019 challenge's task 4 used the Top-1 ranked model [19] of DCASE 2018 challenge's task 4, which proposed Mean Teacher method for SED; however, this baseline was designed with smaller architecture. Thus, we designed our baseline model based on originally Top-1 ranked model in DCASE 2018. Our baseline consists of feature extractor F and classifier F_t which are trained using the strongly labeled synthetic data and weakly labeled real data without Mean Teacher algorithm. The baseline showed 24.15% event-based F1 score. For the comparisons, the official results of various SED frameworks submitted for the DCASE 2019 challenge's task 4 are shown in Table 1.

With our baseline model, we applied the adversarial learning method in two ways. The domain classifier D predicted real or synthetic on whole feature map (*Adv.whole*) and on features in each time frame (*Adv.time*). Both adversarial learning approaches improved the performance as shown in Table 1. These approaches reduced domain gap between synthetic and real feature distributions, thsu we could improve performance since the validation set was also from the real audio clips. The *Adv.time* and *Adv.whole* achieved 31.33% and 30.65%, respectively. The *Adv.time* method showed better performance than the *Adv.whole* since the architecture tries to predict multi-label in each time frame. We also performed pseudo-labeling method using tri-training procedure. The tri-training method achieved 30.23%. Tri-training method showed better performance than the adversarial learning in evaluating segment-based F1 scores. Finally, we evaluated the combined architecture: adversarial learning with the tri-training. When we trained two labelers in tri-training, we also trained domain classifier simultaneously for adversarial learning. After training two labelers with adversarial learning, more confident labelers for predicting sound event classes on real dataset were obtained. Then, we assigned pseudo-label to weakly labeled and unlabeled samples based on two labelers. We trained the final classifier with labeled samples and pseudo-labeled samples by tri-training scheme. In combining the adversarial learning with tri-training method, we considered the previous two approaches: *Adv.whole* and *Adv.time*. The tri-training method combined with *Adv.whole* approach showed 32.64% event-based F1 score and the tri-training method combined with *Adv.time* achieved 35.10%. *Adv.time*+Tri-training achieved the highest event-based F1 score of our models. The tri-training method achieved 62.86% segment-based F1 score and it is better than *Adv.time*+Tri-training. We think that adversarial learning contributes more to inference exact sound label in time frame while the tri-training contributes more to inference the exact boundary of the sound event.

Table 1: The event based macro F1 scores and segment based macro F1 scores of proposed methods on validation dataset in DCASE 2019 challenge's task 4

Model	**Macro F1 (%)**	
	Event-based	Segment-based
Wang_YSU_task4_1	19.4%	-
Kong_SURREY_task4_1	21.3%	-
Wang_NUDT_task4_3	22.4%	-
DCASE 2019 baseline [22]	23.7%	55.2%
Rakowski_SRPOL_task4_1	24.3%	-
mishima_NEC_task4_4	24.7%	-
Lee_KNU_task4_3	26.7%	-
bolun_NWPU_taks4_2	31.9%	-
Kothinti_JHU_task4_1	34.6%	-
ZYL_UESTC_task4_2	35.6%	-
Kiyokawa_NEC_task4_4	36.1 %	-
PELLEGRINI_IRIT_task4_1	39.9%	-
Lim_ETRI_task4_4	40.9 %	-
Shi_FRDC_task4_2	42.5%	-
Yan_USTC_task4_4	42.6 %	-
Delphin_OL_task4_2	43.6%	-
Lin_ICT_task4_3	**45.3**%	-
Our baseline	24.15%	57.70%
Adv.whole	30.65%	59.06%
Adv.time	31.33%	59.26%
Tri-training	30.23%	**62.86**%
Adv.whole + Tri-training	32.64%	60.48%
Adv.time + Tri-training	**35.10**%	60.67%

5. CONCLUSION

In this paper, we consider the semi-supervised learning framework for weakly labeled SED problem for the DCASE 2019 challenge's task4 by combining both tri-training and adversarial learning. The entire dataset consists of the synthetic data with the strong label (sound event labels with boundaries) and real data with weakly labeled (sound event label) and unlabeled dataset. We reduce domain gap between strongly labeled synthetic dataset and weakly labeled or unlabeled real dataset to train networks to learn domain-invariant feature for preventing degradation of performance. Also, we utilize pseudo labeled samples based on confident multiple labelers trained by labeled samples. Then, networks learn the discriminative representation of the unlabeled dataset. The tri-training method combined with adversarial learning on each time frame shows a considerable performance improvement over the baseline model.

6. REFERENCES

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

[2] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *Proc. DCASE, 2018*, 2018.

[3] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 151–155.

[4] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 771–775.

[5] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proceedings of the IEEE European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[7] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[8] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6440–6444.

[9] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time–frequency representations for audio scene classification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 1, pp. 142–153, 2015.

[10] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[11] N. Almaadeed, M. Asim, S. Al-Maadeed, A. Bouridane, and A. Beghdadi, "Automatic detection and classification of audio events for road surveillance applications," *Sensor Signal and Information Processing (SSIP)*, vol. 18, no. 6, 2018.

[12] M. Fan, W. Wang, P. Dong, L. Han, R. Wang, and G. Li, "Cross-media retrieval by learning rich semantic embeddings of multimedia," in *Proceedings of the ACM international conference on Multimedia (ACMMM)*, 2017, pp. 1698–1706.

[13] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the ACM international conference on Multimedia (ACMMM)*, ser. MM '10, 2010, pp. 251–260.

[14] https://store.google.com/gb/product/nest_cam_outdoor.

[15] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 559–563.

[16] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[17] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 171–175.

[18] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proceedings of the ACM international conference on Multimedia (ACMMM)*, 2016, pp. 1038–1047.

[19] L. JiaKai, "Mean teacher convolution system for DCASE 2018 task 4," *Detection and Classification of Acoustic Scenes and Events*, 2018.

[20] http://dcase.community/challenge2018/.

[21] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2017.

[22] http://dcase.community/challenge2019/.

[23] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

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

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International Conference on Machine Learning*, 2015, pp. 2067–2075.

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

[27] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.