SOUND EVENT DETECTION BY CONSISTENCY TRAINING AND PSEUDO-LABELING WITH FEATURE-PYRAMID CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Technical Report

Chih-Yuan Koh[1], You-Siang Chen[1], Shang-En Li[2], Yi-Wen Liu[1], Jen-Tzung Chien[2], Mingsian R. Bai[1]

[1] National Tsing Hua University, Hsinchu 30013, Taiwan,
jimmy133719@gapp.nthu.edu.tw, s108033851@m108.nthu.edu.tw,
ywliu@ee.nthu.edu.tw, msbai@pme.nthu.edu.tw
[2] National Chiao Tung University, Hsinchu 30010, Taiwan,
{w0860239.cm08g, jtchien}@nctu.edu.tw

ABSTRACT

A event detection system for DCASE 2020 task 4 [1] is presented. To efficiently utilize large amount of unlabeled in-domain data, three semi-supervised learning strategies are applied: 1) interpolation consistency training (ICT), 2) shift consistency training (SCT), 3) weakly pseudo-labeling. In addition, we propose FP-CRNN, a convolutional recurrent neural network which contains feature-pyramid components and is based on the provided baseline [2]. In terms of event-based F-measure, these approaches outperform the baseline, at 34.8%, by a large margin, with an F-measure of 48.4% for the baseline network which is trained with the combination of all three strategies and 49.6% for FP-CRNN with the same training strategies.

***Index Terms*—** Sound event detection, weakly supervised learning, semi-supervised learning, CRNN, feature pyramid

1. INTRODUCTION

The goal of sound event detection is to classify acoustic events and find out the event boundaries in an audio clip. Applications of sound event detection include smart home [3], health monitoring systems [4], surveillance [5], and multimedia retrieval [6, 7]. With the growing attention in this field, Detection and Classification of Acoustic Scenes and Events (DCASE), a series of challenges evaluating sound detection and classification systems, has been held since 2013. Task 4 aims to explore the possibility to train sound event detection systems with a large amount of unlabeled data, weakly labeled data which does not provide onsets and offsets of events, and synthetic strongly labeled data. This year, participants are also encouraged to apply sound separation to improve the sound event detection system, with the expectation that this step can separate overlapping sound events and extract foreground sound events from background sound events.

The rest of this paper is organized as follows. Section 2 describes the dataset and audio preprocessing. In Section 3, we introduce three semi-supervised learning strategies and propose a network architecture. Experimental results are shown in Section 4. Conclusions are given in Section 6.

2. DATASET

2.1. DESED

The dataset of DCASE 2020, domestic environment sound event detection (DESED) [8], is comprised of 10-sec audio clips that were either recorded in a domestic environment or synthesized with isolated sound events and backgrounds to simulate a domestic environment. Each audio clip contains at least one sound event corresponding to one of the 10 classes. For real soundscapes, data can be divided into 4 subsets, weakly labeled (1578 clips), unlabeled in-domain (14412 clips), validation (1168 clips), and evaluation. All of them are sampled at 44100 Hz. Weakly labeled data only contain labels of events in audio clips but do not provide time boundaries of events. Unlabeled in-domain data do not supply any labels, but they are ensured to be in the same domain as labeled data. Both validation and evaluation data have complete annotations, and the difference between them is their usage. For synthetic soundscapes, they are only divided into training and evaluation. All of synthetic soundscapes are sampled at 16000 Hz and contain complete annotations.

2.2. Audio preprocessing

Instead of directly feeding audio clips into a neural network, log-Mel spectrograms from audio clips are extracted as inputs. To generate spectrograms, we follow the specification in baseline [2]. Audio clips are resampled at 16000 Hz. Window size, hop length, maximum frequency and number of Mel bins are 2048, 255, 8000, and 128, respectively. We normalize extracted spectrograms along their frequency axis. Consequently, 628×128 spectrograms would be input features for a neural network.

3. METHODS

In this section, we briefly illustrate the method of baseline in 3.1. Section 3.2–3.4 describe three semi-supervised learning methods. The proposed neural network architecture, FP-CRNN, is introduced in 3.5. Finally, 3.6 describes the post-processing.

3.1. Baseline

The baseline network architecture is formed as a convolutional recurrent neural network (CRNN) [9], which consists of 7 layers of

CNN blocks, 2 layers of bidirectional gated recurrent unit (GRU) [10] cells, and an attention part for producing outputs. A CNN block is comprised of batch normalization, 2D convolution, and gated linear unit (GLU) [11] as the activation function. Each CNN block is followed by average pooling. It is worth noting that pooling along the time axis is not applied after all CNN blocks. By doing so, time resolution of features is remained, which is beneficial to promote frame-level predictions. Both CNN blocks and GRU cells apply 50% dropout during training. The attention part contains two dense layers followed by sigmoid and softmax, respectively. The sigmoid yields frame-level predictions for final sound event detection, whereas the softmax outputs are weighted summed with the sigmoid outputs to generate clip-level predictions which are used for training with weakly labeled ground truth.

To utilize the large amount of unlabeled data, the mean-teacher approach [12] is applied. The mean-teacher approach contains a student model and a teacher model. The concept of this approach is to encourage two models to produce close predictions when adding different noise to the input. In terms of implementation, the two models share the same network architecture, but use different weights. The weights of the student model are learned from training, while the weights of the teacher model are updated as an exponential moving average of the student's weights. To carry out the concept, the mean squared error between the outputs of the student model and the teacher model is added into loss function. Since we expect to use this mechanism when the model's accuracy achieves a certain level, a ramp-up function is applied as consistency weight for these terms. The loss function is given as follows,

$$L_{\text{baseline}} = L_{w,\text{BCE}} + L_{s,\text{BCE}} + w(t)(L_{w,\text{MSE}} + L_{s,\text{MSE}}), \quad (1)$$

where

$$w(t) = \exp\left[-5\left(1 - \frac{t^2}{T}\right)\right] \quad (2)$$

is a ramp-up function and the subscripts w (weak) and s (strong) denote clip-level outputs and frame-level outputs, respectively. Subscript BCE and MSE denote binary cross-entropy loss and mean square error, respectively. In the ramp-up function $w(t)$, t denotes the current iteration of training, and T denotes the ramp-up length which is set to 50 epochs in our implementation.

3.2. Interpolation consistency training

We draw on the ideas of a prior work [13] which applied a state-of-the-art semi-supervised learning method, called interpolation consistency training (ICT) [14]. ICT encourages the prediction at an interpolation of unlabeled points to be consistent with the interpolation of predictions at those points, which can be shown in the following equation,

$$f_\theta(\lambda u_j + (1-\lambda)u_k) \approx \lambda f_{\theta'}(u_j) + (1-\lambda) f_{\theta'}(u_k) \quad (3)$$

where f_θ and $f_{\theta'}$ denote a student model and a teacher model, respectively. In practice, λ is randomly sampled from the Beta distribution. It is intuitive that learning from interpolation samples can help the model discriminate samples that are ambiguous between two classes. Implementation of [13] replaces all input samples with interpolation samples and calculates the same loss function as the baseline. However, we find that original input samples can stabilize the model performance during training. Hence, our final loss is the sum of the baseline loss and the loss with interpolation samples as the model inputs.

3.3. Shift consistency training

Inspired by ICT, we came up with an idea called shift consistency training (SCT), which also applies consistency regularization. SCT encourages the prediction of time-shifted inputs to be consistent with time-shifted prediction. The intuition of this method is that model can learn shift invariance with consistency regularization, and this may solve the problem mentioned in [8] that sound events' positions within the clip has a large impact on detection performance for the long sound event classes. Therefore, SCT allows model to learn better on temporal localization of sound events. Moreover, shifting along the frequency axis has also been attempted, and it further improves the model's performance. The reason why frequency shift helps may be due to an increase in the diversity of data. The loss of SCT is yielded by,

$$L_{\text{SCT}} = L_{\text{baseline}} + L_{\text{shift}}, \quad (4)$$

where

$$L_{\text{shift}} = L_{wf,\text{BCE}} + L_{sf,\text{BCE}} + L_{st,\text{BCE}} + w(t)L_{st,\text{MSE}} \quad (5)$$

is the sum of all loss terms related to shift. wf, sf, and st denote clip-level outputs with frequency shift, frame-level outputs with frequency shift, and frame-level outputs with time shift, respectively. Input data without shift are also used during training for the same reason as ICT.

3.4. Weakly pseudo-labeling

Though the goal of this task is only to achieve higher accuracy in sound event detection, we also keep track to the performance of audio tagging problem which is the evaluation of clip-level prediction. To our surprise, the performance of sound event detection is difficult to be improved, but that of audio tagging can be easily improved by changing CRNN to deeper network architecture. Due to more convincing clip-level predictions, these deeper models can be employed to generate reliable weak pseudo-labels.

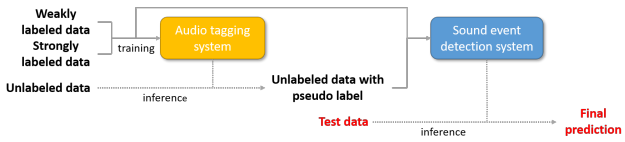

Figure 1: The proposed weakly pseudo-labeling method.

As illustrated in Fig. 1, this weakly pseudo-labeling strategy can be viewed as two-stage learning. The audio tagging system refers to the deeper model, while the sound event detection system is still CRNN. With these weak pseudo-labels, the task is converted into pure weakly supervised learning.

3.5. Feature-pyramid CRNN

Besides exploring the method to use data efficiently, we also committed to study the network architecture itself. Considering different duration of each sound event class, the system needs to be invariant against the scale of patterns. Indeed, CNN with pooling layers already has the advantage of scale-invariance, but applying a feature-pyramid component which utilizes multi-scale features has been verified to further enhance this advantage. In [15], feature-pyramid networks are applied to solve the task of object detection.

The predictions that are inferred from different scale features are aggregated to produce final object detection outputs. Such a concept provides a significant improvement.

Sound event detection can be viewed as object detection in the audio domain, and hence a feature-pyramid component might also be useful. This idea has already been applied; in [16], the second last layer in CNN part was pooled with different sizes, and these output feature maps with different scales were then upsampled and combined with that of the last layer. Comparing with [16], two more CNN blocks and average pooling layers are added to the last layer of CNN part instead of using previous layers' outputs in our work. Furthermore, we do not aggregate these different scale features until passing through bidirectional GRUs. Then, 1×1 convolution is applied after concatenation of different scale features to smooth the combined features and reduce their dimension. Fig. 2 shows the network architecture with a feature-pyramid component which is named as FP-CRNN.

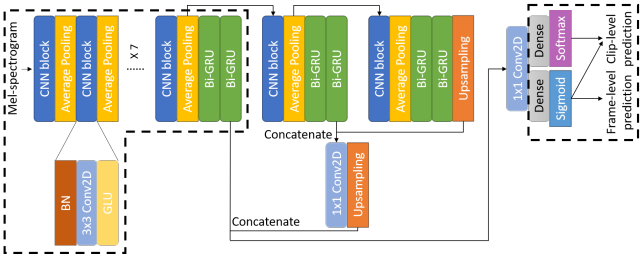

Figure 2: The architecture of FP-CRNN. Dotted portion is the baseline CRNN, and the remaining part is the proposed feature-pyramid component.

During evaluating FP-CRNN, we found that the performance of audio tagging was improved by a large margin. F-measure increases about 25% when a feature-pyramid component is added. To fully exploit this advantage, a binary mask is generated from clip-level outputs. The outputs which are higher than the threshold would be set to 1, otherwise they would be set to 0. By applying the mask to frame-level outputs, the model produces fewer false-positive predictions since the classes with low confidence are eliminated.

3.6. Adaptive post-processing

The frame-level outputs of the model may be non-consecutive, and this would result in producing too many sound event predictions with very short duration. Hence, the common solution is to apply median filters to smooth these outputs. Given that each sound event class has its own duration, using median filters with fixed window length may not be enough. According to the statistics of events duration in [17], we divide sound event classes into two group, background sounds and impulsive sounds. Sound event classes with long duration belong to background sounds, while those with short duration belong to impulsive sounds. Empirically, applying median filters with two different window lengths for the two groups can give a better smoothness result.

4. EXPERIMENTS

4.1. Experimental setup

For the model shown in Fig. 2, the specification of the basline's CRNN part is in [2], and the detail of remaining part is shown in Table 1. For ICT, α in Beta(α, α) is the only parameter needed to be tuned. We set α to 1 and 2 for labeled data and unlabeled data, respectively. For SCT, the amount of time shift and frequency shift are sampled from uniform distribution between $\pm$2s and $\pm$4 mel bins, respectively. A pretrained ResNet18 [18] is fine-tuned to infer weak pseudo-labels from unlabeled data. Window length of median filters for background sounds and impulsive sounds are 2.7s and 0.45s, respectively. As for the parameters of training, they are the same as the provided baseline system [2].

Table 1: The specification of FP-CRNN besides the baseline part.

Component	Description
CNN block	128 channels 3×3 Conv2D
Average pooling	2×1 (time/frequency)
1×1 Conv2D	256 channels
Upsampling	Bilinear upsampling

4.2. Evaluation of semi-supervised learning strategies

Table 2: The performance of models using different semi-supervised learning strategies and post-processing. **Pseudo** and **Post** denote weakly pseudo-labeling and adaptive post-processing, respectively.

Strategies				Results	
ICT	**SCT**	**Pseudo**	**Post**	**F1 (%)**	**PSDS (%)**
—	—	—	—	34.8	60.0
✓				40.6	65.0
	✓			40.4	62.8
		✓		38.8	63.1
			✓	37.4	63.3
✓	✓	✓	✓	**47.2**	**67.3**

The proposed semi-supervised strategies and post-processing approach are evaluated with the validation set under event-based measurement. Macro F1 score is the primary metric, and polyphonic sound detection score (PSDS) [19] acts as the auxiliary metric. The result of baseline CRNN models with different strategies is shown in Table 2. The first row without any ticks represents the baseline. All of the proposed strategies boost up the performance. Among them, ICT and SCT appear to be more effective. We can say that these strategies work as data augmentation techniques. By increasing the diversity of input data, the model becomes more robust. In addition, these strategies are mutually compatible, which further improves the performance by at least 7% comparing to using only one strategy.

4.3. Evaluation of different network architectures

In Table 3, different network architectures are evaluated with or without applying the combination of strategies. The result shows

Table 3: The performance of different network architectures. **SED** and **AT** represent sound event detection and audio tagging, respectively.

Methods		**SED**		**AT**
Model	**Strategies**	**F1 (%)**	**PSDS (%)**	**F1 (%)**
CRNN		34.8	60.0	49.7
CRNN	✓	**47.2**	67.3	44.7
FP-CRNN		39.1	64.7	75.6
FP-CRNN	✓	45.7	**69.0**	**78.0**

that the audio tagging performance is significantly improved when adding the feature-pyramid component. By further exploiting this advantage, false-positive frame-level predictions are reduced, which improves the performance of sound event detection. It is interesting that FP-CRNN has a relative small improvement when used in combination with all the strategies. Our explanation is that FP-CRNN itself has slightly better performance on audio tagging than the deeper architecture used in weekly pseudo-labeling approach, so the strategies in Table 2 are of limited help to FP-CRNN.

4.4. Model ensemble

To further improve the performance of our work, we apply model ensemble to both network architectures with different max consistency weights used in the mean-teacher approach, ICT, and SCT. For CRNN, the outputs of models with a maximum consistency weight of 1.5, 2.0, and 2.5 are averaged to generate final predictions. For FP-CRNN, we ensemble the models with a maximum consistency weight of 2.0, 2.25, 2.5, 2.75, and 3.0 for final inference.

Table 4: The performance of model ensemble

Methods		**Results**	
Model	**Post**	**F1 (%)**	**PSDS(%)**
Baseline		34.8	60.0
Ensemble CRNN		46.4	68.4
Ensemble CRNN	✓	48.4	70.1
Ensemble FP-CRNN		48.0	70.1
Ensemble FP-CRNN	✓	**49.6**	**70.9**

As shown in Table 4, model ensemble improves the performance by about 2% comparing to the best performance in Table 3. Though adaptive post-processing is verified to obtain better performance on validation set, we are concerned that it may not work on other datasets. The analysis in [8] shows that the model using fixed length median filter does better on detecting a long sound event class at the end of an audio clip. As a result, we also use the models without using adaptive post-processing to infer predictions of evaluation set. Four ensemble systems in Table 4 are submitted to DCASE 2020 task 4. The submitted models' names from top to bottom are Koh_NTHU_task4_SED_1, Koh_NTHU_task4_SED_2, Koh_NTHU_task4_SED_3, and Koh_NTHU_task4_SED_4.

5. CONCLUSION

In this work, we explore the possibility to utilize unlabeled data and also propose a new network architecture. Three semi-supervised learning strategies have been applied under the same network. ICT helps the models learn from ambiguous samples, and SCT assists the models in learning temporal information of sound events. Both of them can be viewed as data augmentation techniques that increase the diversity of data. Weakly pseudo-labeling transforms unlabeled data into somewhat reliable weakly-labeled data. FP-CRNN utilizes different scales of features and is shown to improve audio tagging performance significantly. Adaptive post-processing is also applied to smooth model outputs. Our best model achieves 49.6% on the validation set, improving the performance by about 15% from the baseline.

6. ACKNOWLEDGMENT

This work is supported by U-MEDIA Communications. We are also grateful to DCASE2020 team for holding this challenge.

7. REFERENCES

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

[2] https://github.com/turpaultn/dcase20_task4/tree/public_branch/baseline.

[3] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 81–94, 2016.

[4] Y. Zigel, D. Litvak, and I. Gannot*, "A method for automatic fall detection of elderly people using floor vibrations and sound—proof of concept on human mimicking doll falls," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.

[5] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, 2005, pp. 158–161.

[6] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia*, vol. 3, no. 3, pp. 27–36, 1996.

[7] Q. Jin, P. Schulam, S. Rawat, S. Burger, D. Ding, and F. Metze, "Event-based video retrieval using audio," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[8] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 86–90.

[9] E. Çakır, 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, no. 6, pp. 1291–1303, 2017.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[11] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[12] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, 2017, pp. 1195–1204.

[13] Z. Shi, L. Liu, H. Lin, R. Liu, and A. Shi, "Hodgepodge: Sound event detection based on ensemble of semi-supervised learning methods," *arXiv preprint arXiv:1907.07398*, 2019.

[14] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *arXiv preprint arXiv:1903.03825*, 2019.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[16] J. Yan, Y. Song, W. Guo, L. Dai, I. McLoughlin, and L. Chen, "A region based attention method for weakly supervised sound event detection and classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 755–759.

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

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] 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.