IMPROVED STUDENT MODEL TRAINING FOR ACOUSTIC EVENT DETECTION MODELS

Anthea Cheung, Qingming Tang, Chieh-chi Kao, Ming Sun, Chao Wang

Amazon Alexa
101 Main St, Cambridge, MA 02142, USA
{antheach, qmtang, chiehchi, mingsun, wngcha}@amazon.com

ABSTRACT

We introduce several novel knowledge distillation techniques for training a single shallow model of three recurrent layers for acoustic event detection (AED). These techniques allow us to train a generic shallow student model without many convolutional layers, ensembling, or custom modules. Gradual incorporation of pseudolabeled data, using strong and weak pseudolabels to train our student model, event masking in the loss function, and a custom SpecAugment procedure with event-dependent time masking all contribute to a strong event-based F1-score of 42.7%, which matches the top submission score, compared to 34.7% when training with a generic knowledge distillation method. For comparison to state-of-the-art performance, we use the ensemble model of the top submission in the challenge as a fixed teacher model.

Index Terms— Acoustic event detection, knowledge distillation, pseudolabeling, SpecAugment

1. INTRODUCTION

Acoustic event detection (AED) is the task of predicting sound events and their time boundaries. It is an emerging area of research as the ability to correctly detect the start and end times of sound events has many useful practical applications in media indexing and retrieval [1], surveillance [2], enhancing smart home devices' ability to interpret the acoustic environment of the home [3]. Compared to audio tagging tasks, acoustic event detection remains a challenging area of research due to the difficulty of obtaining high-quality annotated clips which contain labels of onset and offset times.

We focused on task four of the 2019 edition of DCASE as it is solely focused on AED and provides a test dataset that can be compared with the top performances in the challenge [4]. In this task, the top-performing submissions are ensembles of models [5, 6] or comprised of multiple layers of convolutional layers [7, 6] and modules with custom architecture [5], which often consume significant memory and are less practical to use in resource-constrained mobile devices. Our focus is on using knowledge distillation techniques to achieve a single shallow model of three recurrent layers with a small degradation in accuracy. We used the ensemble model of the top submission in the challenge as a fixed teacher model.

Recent results on noisy student training explored promising techniques for an automatic speech recognition (ASR) task [8]. Firstly, they tried gradually introducing harder samples to the training of the student model by applying a score on each utterance-transcript pair and lowering the cutoff score for each generation of training. Curriculum learning applies a similar concept in slowly expanding the training set for the student model. Secondly, they performed SpecAugment [9] and increased the time masking length to produce harder samples for the student model. To our knowledge, neither techniques have been explored for AED before. We apply gradual incorporation of pseudolabeled data, strong and weak pseudolabels to train our student model, and a custom SpecAugment procedure with event-dependent time masking to achieve a strong event-based F1-score of 42.7%.

2. RELATED WORK

Recently, the use of deep learning models with convolutional neural network (CNN) [5, 10] and convolutional recurrent network network (CRNN) [4] architectures have yielded the best performance in AED tasks. More recently, custom solutions such as disentangled features [11] and independent component [12] modules have been added on top of CNN or CRNN architectures to further refine performance. As strongly-labeled AED datasets are relatively small, semi-supervised methods are used to take advantage of unlabeled and weakly-labeled sets, either by only using weak predictions [5, 13] or both strong and weak predictions [14, 4]. Knowledge distillation [15] has been studied for AED using only weak labels [16] or using weak and strong labels in two stages [17]. Mean-teacher models [18] use a similar concept in applying a consistency loss to student and teacher models with the same architecture.

The importance of different time scales of the present events are evident in the post-processing steps of several systems for AED [7, 19]. These improvements inspired us to mask the input features and predictions based on the labeled classes. Masking of time and frequency bands is used in SpecAugment [9], but most AED systems only use time and frequency masking not time warping [20, 21]. Partially masking the model outputs during gradient descent have been used for AED [22] and for localization tasks [23]. Curriculum learning [24] is a strategy of gradually adding more difficult samples during training, and has been used to train ASR [8], emotion recognition [25], and translation models [26]. Tonami et al [27] studied curriculum for AED but ranked samples' difficulty based on the presence of labeled classes.

3. METHODOLOGY

The DCASE 2019 task 4 dataset consists of 10 different sound events from domestic environments. The training dataset contains synthetic clips strongly-labeled with onset and offset times, weakly-labeled real recordings that contain event labels but no onset and offset times, and a large unlabeled dataset of real recordings. The validation and public test sets are both strongly-labeled datasets with real recordings.

3.1. Our model

Our student model $\mathcal{S}$ was designed to be a simple recurrent neural network (RNN) model, achieving close to state-of-the-art performance solely relying on data augmentation and knowledge distillation techniques without requiring hand-crafted features or custom architectures. We provided as input $\boldsymbol{X} \in R^{d \times T}$ log-mel features with dimension $d = 20$ and time steps $T = 500$. Our model uses a three-layer uni-directional LSTM architecture with $h = 256$ nodes in each layer to produce an embedding $g(\boldsymbol{X}) = \boldsymbol{H} \in R^{h \times T}$. To generate the onset and offset times of each predicted event c in the clip, we obtained a frame-level prediction $\boldsymbol{f} \in R^{C \times T}$ by passing $\boldsymbol{H}$ through a fully-connected layer $\boldsymbol{W}$ with C output classes and a sigmoid activation function.

We then used an attention-pooling mechanism to generate the audio tagging predictions. For each class c, the attention weights $\boldsymbol{z}_c \in R^T$ are obtained by:

$$\boldsymbol{z}_c = \frac{exp(\boldsymbol{a}_c \boldsymbol{H} + \boldsymbol{b}_c)}{\sum_{k=1}^{C} exp(\boldsymbol{a}_k \boldsymbol{H} + \boldsymbol{b}_k)}, \tag{1}$$

where $a_k, b_k \in R^h$ are the class weights and bias vectors for class k. The clip level audio tagging outputs are obtained by normalizing the frame outputs $\boldsymbol{f}_c$ with the attention weights $\boldsymbol{z}$. All frame predictions for events with a clip level prediction below the threshold of 0.5 were set to zero so that only clips above the threshold also had positive frame predictions.

3.2. Use of pseudolabels

We aim to understand the relative benefits of using pseudolabels, and whether or not progressively incorporating easier or harder samples yield better results. To that end, we used the teacher model $\mathcal{T}$ from the top-performing submission in the DCASE 2019 task 4 challenge [5], which is an ensemble of six CNNs with an attention pooling layer. The audio tagging and detection results from teacher are generated for the unlabeled set without applying post-processing steps, and used as targets for training the student model. After the end of an epoch, the pseudolabeled samples can be evaluated by the student model. At each generation of training, we deemed samples whose student model predictions more closely match the teacher model's soft targets as easier samples. We used the following heuristic to score the difficulty of each sample $\boldsymbol{X}$ using the weak predictions $\boldsymbol{t}, \boldsymbol{s} \in R^C$ of the teacher and student models, respectively:

$$\mu(\boldsymbol{t}, \boldsymbol{s}) = max_c(|\boldsymbol{t}_c - \boldsymbol{s}_c|), \tag{2}$$

which is the maximum difference in the teacher and student scores across all C classes. The maximum difference rather than mean is chosen to ensure that the score discrepancies across all events are below $\mu(\boldsymbol{t}, \boldsymbol{s})$.

3.3. Customized SpecAug procedure

Although the standard SpecAug [9] procedure applies a fixed length of time masking to each clip, the average duration of different sound events vary greatly. For example, the duration of dishes clanging is much shorter than that of vacuuming, so a model for vacuum sound should be robust to longer time masks compared to a model for dishes. Applying this principle, we devised a customized procedure that varies the length of the time mask. For clips from the unlabeled dataset, we used the weak soft targets generated by the teacher as labels; otherwise we use the original labels. We added random noise $\varepsilon_c \sim N(0, 1\mathrm{e}-6)$ for each event category to get noisy labels $\widetilde{\boldsymbol{Y}_c} = \boldsymbol{Y}_c + \varepsilon$. For the top K events in $\widetilde{\boldsymbol{Y}_c}$, we apply a time mask to $\boldsymbol{X}$ with the length given by $\gamma \cdot l_c$, where l_c is the median frame length of event c. We also masked frequency bands F times with fixed length L_f in the same way as the standard SpecAug procedure. Our tunable hyperparameters are F and L_f for frequency masking and K and γ for time masking.

4. TRAINING

We applied the same procedure as that of the original teacher model submission to produce 64-dimensional log-mel filterbank features. A window length of 40 ms, hop length of 20 ms, and 2048 number of fft components were used to produce 500 frames for each audio clip. For the student model, we used 20-dimensional log-mel filterbank features with the same window and hop lengths to produce 500 frames for each clip.

4.1. Loss function

Our dataset consists of: 1) a strongly-labeled synthetic dataset $\mathcal{L}^S$; 2) a weakly-labeled dataset $\mathcal{L}^W$; and 3) an unlabeled dataset $\mathcal{U}$. For each training sample $\boldsymbol{X}$ of the strongly-labeled synthetic dataset, we have strong audio detection labels $\boldsymbol{Y}^s$, and inferred weak labels $\boldsymbol{Y}^w$, where each class has $\boldsymbol{Y}^w_c = \max_t(\boldsymbol{Y}^s_{c,t})$, while the weakly-labeled dataset only has weak labels $\boldsymbol{Y}^w$. For each sample in the unlabeled dataset, we denote the teacher strong and weak predictions by $\boldsymbol{t}^s$ and $\boldsymbol{t}^w$, respectively, and the student strong and weak predictions by $\boldsymbol{s}^s$ and $\boldsymbol{s}^w$. During training, the loss function is composed of the weak loss, strong loss, and unlabeled loss with hyperparameters λ_1 and λ_2 which are weights for the clip and frame level losses:

$$\mathcal{J} = J^w + J^s(\lambda_1, \lambda_2, \boldsymbol{M}) + J^u(\lambda_1, \lambda_2, \boldsymbol{M}) \tag{3}$$

where the weak loss J^w is the binary cross-entropy loss

$$J^w = \frac{1}{|\mathcal{L}^w|} \sum_{X \in \mathcal{L}^w} (\boldsymbol{Y}^w \log(\hat{\boldsymbol{Y}^w}) + (\boldsymbol{1} - \boldsymbol{Y}^w) \log(\boldsymbol{1} - \hat{\boldsymbol{Y}^w})) \tag{4}$$

and the strong loss is

$$J^s = \frac{\lambda_1}{|\mathcal{L}^s|} \sum_{\boldsymbol{X} \in \mathcal{L}^s} (\boldsymbol{Y}^w \log(\hat{\boldsymbol{Y}^w}) + (\boldsymbol{1} - \boldsymbol{Y}^w) \log(\boldsymbol{1} - \hat{\boldsymbol{Y}^w}))$$
$$+ \frac{\lambda_2}{|\mathcal{L}^s|} \sum_{\boldsymbol{X} \in \mathcal{L}^s} (\boldsymbol{Y}^s \log(\boldsymbol{M} \odot \hat{\boldsymbol{Y}^s}) + \boldsymbol{1} - \boldsymbol{Y}^s) \log(\boldsymbol{1} - \boldsymbol{M} \odot \hat{\boldsymbol{Y}^s})) \tag{5}$$

The loss J^s for strongly-labeled samples is a weighted sum of the clip-level and framewise binary cross-entropy losses. Since the labels in the dataset are sparse, the average framewise loss can be quite inefficient as the composition of the loss may be dominated by the cross-entropy loss of negative frames. Thus, we compared the results of three different types of masking: 1) no masking; 2) event masking; and 3) segment masking.

In the case of no masking, $\boldsymbol{M} \in R^{C \times T}$ is simply a matrix of ones. For event masking, we take the Hadamard product of the predictions $\hat{\boldsymbol{Y}^s}$ and the mask

$$\boldsymbol{M}_{ij} = \begin{cases} 1, & Y_i = 1 \\ 0, & Y_i = 0 \end{cases}, \tag{6}$$

Experiment	Curriculum	Pseudolabels	Masking	SpecAug	Best val F1	Best test F1	Mean±sd val F1	Mean±sd test F1
Lin_ICT_3	-	-	-	-	45.3	42.7	-	-
Lin_ICT_2	-	-	-	-	44.0	40.9	-	-
EF+SW+EM+CS	Easier	Strong+weak	Event	Custom	41.6	42.7	40.7 ± 0.6	41.3 ± 0.9
SW+EM+CS	All	Strong+weak	Event	Custom	41.3	42.5	40.3 ± 0.7	41.0 ± 1.1
HF+SW+EM+CS	Harder	Strong+weak	Event	Custom	40.7	42.2	40.3 ± 0.8	40.3± 1.2
SW+NM+NS	All	Strong+weak	None	None	34.1	34.7	33.1±0.7	33.5±0.9

Table 1: Comparison of pseudolabel scheduling (easier first, harder first, or adding all at once).

Experiment	Curriculum	Pseudolabels	Masking	SpecAug	Best val F1	Best test F1	Mean±sd val F1	Mean±sd test F1
EF+SW+EM+CS	Easier	Strong+weak	Event	Custom	41.6	42.7	40.7 ± 0.6	41.3 ± 0.9
EF+SW+EM+SS	Easier	Strong+weak	Event	Standard	40.7	41.6	40.0± 0.4	40.9± 0.6
EF+SW+EM+NS	Easier	Strong+weak	Event	None	40.2	41.2	39.5 ± 0.4	39.9 ± 0.7
EF+SW+EM+CS	Easier	Strong+weak	Event	Custom	41.6	42.7	40.7 ± 0.6	41.3 ± 0.9
EF+SW+SM+CS	Easier	Strong+weak	Segment	Custom	39.9	40.0	39.2± 0.4	39.1 ± 0.5
EF+SW+NM+CS	Easier	Strong+weak	None	Custom	35.7	34.5	33.9 ± 0.8	32.9 ± 0.8

Table 2: Comparison of data augmentation methods (custom SpecAug, standard SpecAug, and no augmentation) and different masking schemes.

so that only frames of present events contribute to the framewise loss. In the case of segment masking, we use the mask

$$\boldsymbol{M}_{ij} = \begin{cases} 1, & Y_{i+12} = 1 \text{ or } Y_{i-12} = 1 \text{ or } Y_i = 1 \\ 0, & \text{otherwise} \end{cases}, \qquad (7)$$

that consists of 1's for corresponding onset and offset frames of each event with a 12-frame buffer before and after each segment (0.24 seconds before and after the onset and offset, respectively).

4.2. Post-processing

After obtaining the framewise outputs for the detection task, we applied the same post-processing step as the procedure in the teacher model. A median filter is applied to each event type, with a window size 1/3 the median number of frames for each event type in the synthetic labeled set.

5. RESULTS

We trained the following types of experiments:

1. Adding in all samples compared to adding easier or harder samples of the pseudolabeled dataset first;
2. Applying no data augmentation compared to using standard SpecAug and our custom SpecAug procedure;
3. Applying no masking, event masking, or segment masking to the loss function;
4. Using only weak or weak and strong pseudolabels on the unlabeled dataset.

Each experiment is trained with batch size 16 and learning rate 0.001 on an Adam optimizer with weights $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The macro event-based F1-score on the validation set is computed at the end of each epoch, and the weights of the best epoch is saved. As the number of training steps for the epochs of the experiments for may differ, in experiment 1, we trained each trial for 48,790 training steps and verified that the validation metric has converged for each experiment type.

5.1. Effect of adding pseudolabels in different stages

We compared the effect of adding all pseudolabels at once and adding easier or harder samples of the pseudolabeled dataset first. For all experiments, we trained in generations of five epochs each. In the first generation, only the labeled dataset was added. For our control group experiment, we addded all the pseudolabels; otherwise, we evaluated the difficulty of each pseudolabel by generating the scores as described in section 3.2. All the samples were ranked in order of their scores, where higher scores mean that the samples are harder for the student model as the student predictions are farther away from the teacher predictions. For the next two generations, the bottom 20% and bottom 40% scoring pseudolabel samples were added to the training set when incorporating easier samples first. We reversed this when incorporating harder samples first, i.e. the top 20% and top 40% scoring pseudolabel samples are added in generations two and three, respectively. Starting from the fourth generation, all pseudolabels were added.

For hyperparamter tuning, we tested each experiment type with multiple hyperparameter values for the loss weights $\lambda_1 \in \{0.5, 0.75, 1, 1.5, 2\}$ and $\lambda_2 \in \{0.5, 0.75, 1, 1.5, 2\}$. After finding the best hyperparameters for each experiment type, we repeated the training process with different random initializations to obtain a total of 10 different runs for each configuration.

The results are summarized in Table 1. We compared our results to the top submission in the challenge (Lin_ICT_3) and the top single model by the same team (Lin_ICT_2). Four models are compared: SW indicates that both strong and weak pseudolabels were used, EM indicates event masking, and CS indicates custom SpecAug procedure. The last experiment (SW+NM+NS) is a baseline knowledge distillation result that does not add masking or augmentation. All pseudolabels are added at once in SW+EM+CS, whereas easier samples are added first in EF+SW+EM+CS, and harder samples are added first in HF+SW+EM+CS. The results show that the best performance is attained by adding easier samples first, with a best event-based macro F1 score of 42.7%, on par with the best performing challenge submission. There is a mod-

Experiment	Curriculum	Pseudolabels	Masking	SpecAug	Best val F1	Best test F1	Mean±sd val F1	Mean±sd test F1
EF+SW+EM+CS	Easier	Strong+weak	Event	Custom	41.6	42.7	40.7 ± 0.6	41.3 ± 0.9
EF+W+EM+CS	Easier	Weak only	Event	Custom	28.4	28.6	26.0±1.0	25.6 ± 1.8
EM+CS	N/A	None	Event	Custom	23.4	24.7	21.0 ± 1.6	21.2± 2.0

Table 3: Validation and test F1 scores for using weak, strong, and no pseudolabels.

Comparison	t-statistic	Statistically significant at		
		$\alpha = 0.2$	$\alpha = 0.05$	$\alpha = 0.01$
Easier first vs all	1.413	Y	N	N
Harder first vs all	0.143	N	N	N
Easier vs harder first	1.483	Y	N	N
Cust. SpecAug vs std.	2.802	Y	Y	N
Cust. SpecAug vs none	7.055	Y	Y	Y
Std. SpecAug vs none	2.989	Y	Y	Y
Event mask vs segment	6.800	Y	Y	Y
Segment mask vs none	19.875	Y	Y	Y
Event mask vs none	23.021	Y	Y	Y
Weak pseduolabels vs none	8.361	Y	Y	Y
Strong + weak vs weak	41.954	Y	Y	Y
Stong + weak vs none	36.011	Y	Y	Y

Table 4: Validation and test F1 scores for different pseudolabels.

est positive effect in adding easier samples first, but incorporating harder samples first does not have much benefit.

5.2. Effect of custom SpecAug procedure

Our best performing model is achieved by applying a custom time-masking SpecAug procedure randomly during training. We tried different values for the hyperparameter governing the length of the time masking $\gamma \in 0.25, 0.5, 1.0$. A final value of $\gamma = 0.25$ was fixed as the best time masking length. In the standard SpecAug experiment time masking was applied with fixed length of 16 time frames. For both procedures, we fixed the probability of applying SpecAug at $p = 0.5$ and apply 1 frequency mask of mask length 4 and 2 time masks. No time warping was applied in either procedure.

The effect of the SpecAug experiments are summarized in Table 2. We compared the performance of the overall top performing configuration (EF+SW+EM+CS, adding easier pseudolabels first) with applying standard SpecAug (EF+SW+EM+SS) and no data augmentation (EF+SW+EM+NS). CS, SS, and NS denote custom SpecAug, standard SpecAug, and no SpecAug, respectively. All other hyperparameters were kept fixed in the experiments, and each experiment is repeated to get ten trials with random initialization.

5.3. Effect of masking on the loss function

Additionally, we compared the effect of adding a segment and event masking matrix when computing the loss on strongly-labeled samples, as detailed in Eq 5. A comparison of ten trials for each masking type is shown in Table 2. The best results were achieved using event masking (EM), where only positive events were included in the calculation of the strong loss. Segment masking (SM) is noticeably worse than event masking but still performs much better than no masking (NM), suggesting that masking helps the student model learn which events are most important in the detection output, but focusing only on positive segments is too aggressive compared to simple event masking.

5.4. Effect of adding weak and strong pseudolabels

In our strongest model, both strong and weak predictions were used as pseudolabels on the unlabeled samples. The results are summarized in Table 3, where SW denotes adding both strong and weak pseudolabels and W denotes only adding weak pseudolabels. While adding weak pseudolabels does significantly boost the performance of the student model (EM+CS), the effect is the largest when comparing adding both strong and weak pseudolabels (EF+SW+EM+CS) with adding only weak pseudolabels (EF+W+EM+CS).

6. CONCLUSION

We have demonstrated that several techniques can be used to train a three-layer LSTM model on AED by using soft targets generated by a strong teacher model. In particular, progressively applying pseudolabeled samples, using variable-length time masking in SpecAug augmentation, and applying event masking to the loss function all contribute to a single model with a 42.7% macro event-based F1-score on the test set, matching state of the art performance of 42.7%. For each of the techniques, we perform a t-test on the means of the validation F1 score of two independent samples to test the statistical significance, summarized in Table 4. We find that adding easier samples first in the pseudolabeled dataset is statistically significant at the $\alpha = 0.2$ level, while the other techniques are significant at the $\alpha = 0.05$ level.

These techniques can be applied to AED models outside the teacher-student training context and can be further studied in more detail. Adding pseudolabeled data in different generations can help models learn more difficult samples over time. Further research can fine-tune these techniques in making the task harder in later generations. For example, adjusting time-masking techniques for data augmentation can be helpful for tasks with events of different average time-scales.

7. REFERENCES

[1] 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.

[2] 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.

[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] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, New York City, United States, Oct. 2019. [Online]. Available: https://hal.inria.fr/hal-02160855

[5] L. Lin and X. Wang, "Guided learning convolution system for dcase 2019 task 4," Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, Tech. Rep., June 2019.

[6] Z. Shi, "Hodgepodge: Sound event detection based on ensemble of semi-supervised learning methods," Fujitsu Research and Development Center, Beijing, China, Tech. Rep., June 2019.

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

[8] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1470

[9] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[10] T. K. Chan, C. S. Chin, and Y. Li, "Non-negative matrix factorization-convolutional neural network (NMF-CNN) for sound event detection," *CoRR*, vol. abs/2001.07874, 2020. [Online]. Available: https://arxiv.org/abs/2001.07874

[11] L. Lin, X. Wang, H. Liu, and Y. Qian, "Disentangled feature for weakly supervised multi-class sound event detection," *CoRR*, vol. abs/1905.10091, 2019. [Online]. Available: http://arxiv.org/abs/1905.10091

[12] X. Zheng, Y. Song, J. Yan, L.-R. Dai, I. McLoughlin, and L. Liu, "An effective perturbation based semi-supervised learning method for sound event detection." in *INTERSPEECH*, 2020, pp. 841–845.

[13] S. Adavanne, H. Fayek, and V. Tourbabin, "Sound event classification and detection with weakly labeled data," 2019.

[14] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[16] R. Shi, R. W. M. Ng, and P. Swietojanski, "Teacher-student training for acoustic event detection using audioset," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 875–879.

[17] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Iterative knowledge distillation in r-cnns for weakly-labeled semi-supervised sound event detection," DCASE2018 Challenge, Tech. Rep., September 2018.

[18] 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 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1195–1204.

[19] L. Cances, T. Pellegrini, and P. Guyot, "Multi task learning and post processing optimization for sound event detection," IRIT, Université de Toulouse, CNRS, Toulouse, France, Tech. Rep., June 2019.

[20] H. Phan, L. Pham, P. Koch, N. Duong, I. McLoughlin, and A. Mertins, "Audio event detection and localization with multitask regression network," DCASE2020 Challenge, Tech. Rep., July 2020.

[21] W. Lim, "Specaugment for sound event detection in domestic environments using ensemble of convolutional recurrent neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 129–133.

[22] W. Wang, C.-C. Kao, and C. Wang, "A simple model for detection of rare sound events," in *Proc. Interspeech 2018*, 2018, pp. 1344–1348. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-2338

[23] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," *arXiv preprint arXiv:1905.00268*, 2019.

[24] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML '09*, 2009.

[25] R. Lotfian and C. Busso, "Curriculum learning for speech emotion recognition from crowdsourced labels," vol. 27, no. 4, 2019.

[26] C. Wang, Y. Wu, S. Liu, M. Zhou, and Z. Yang, "Curriculum pre-training for end-to-end speech translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 3728–3738. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.344

[27] N. Tonami, K. Imoto, Y. Okamoto, T. Fukumori, and Y. Yamashita, "Sound event detection based on curriculum learning considering learning difficulty of events," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 875–879.